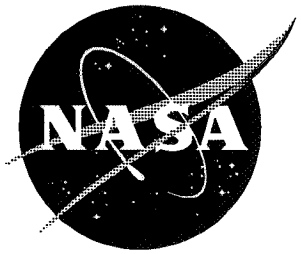


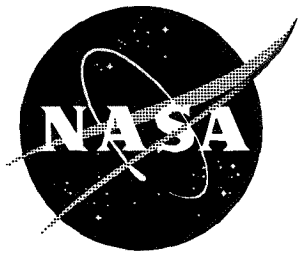
NASA Conference Publication 3339
Part 1

Seventh Copper Mountain Conference on Multigrid Methods

Edited by
N. Duane Melson, Tom A. Manteuffel, Steve F. McCormick, and Craig C. Douglas

Proceedings of a workshop cosponsored by the
National Aeronautics and Space Administration,
Washington, D.C., and the Department of Energy,
Washington, D.C., and held at
Copper Mountain, Colorado
April 2–7, 1995

September 1996



NASA Conference Publication 3339
Part 1

Seventh Copper Mountain Conference on Multigrid Methods

Edited by
N. Duane Melson
Langley Research Center • Hampton, Virginia

Tom A. Manteuffel and Steve F. McCormick
University of Colorado • Boulder, Colorado

Craig C. Douglas
IBM Thomas J. Watson Research Center • Yorktown Heights, New York
Yale University • New Haven, Connecticut

Proceedings of a workshop cosponsored by the
National Aeronautics and Space Administration,
Washington, D.C., and the Department of Energy,
Washington, D.C., and held at
Copper Mountain, Colorado
April 2–7, 1995

National Aeronautics and Space Administration
Langley Research Center • Hampton, Virginia 23681-0001

September 1996

The use of trademarks or names of manufacturers in this report is for accurate reporting and does not constitute an official endorsement, either expressed or implied, of such products or manufacturers by the National Aeronautics and Space Administration.

This publication is available from the following sources:

NASA Center for AeroSpace Information
800 Elkridge Landing Road
Linthicum Heights, MD 21090-2934
(301) 621-0390

National Technical Information Service (NTIS)
5285 Port Royal Road
Springfield, VA 22161-2171
(703) 487-4650

PREFACE

The *Seventh Copper Mountain Conference on Multigrid Methods* was held on April 2–7, 1995, at Copper Mountain, Colorado, and was sponsored by NASA and the Department of Energy. The University of Colorado, Front Range Scientific Computations, Inc., and the Society for Industrial and Applied Mathematics provided organizational support for the conference.

This document is a collection of many of the papers that were presented at the conference and thus represents the conference proceedings. NASA Langley has graciously provided printing of this book so that all of the papers could be presented in a single forum. Each paper was reviewed by a member of the conference organizing committee under the coordination of the editors.

The multigrid discipline continues to expand and mature, as is evident from these proceedings. The vibrancy and diversity in this field are amply expressed in these important papers, and the collection clearly shows the continuing rapid growth of the use of multigrid acceleration techniques.

N. Duane Melson
NASA Langley Research Center

Steve F. McCormick and
Tom A. Manteuffel
University of Colorado at Boulder

Craig Douglas
IBM Thomas J. Watson Research Center
Yale University

ORGANIZING COMMITTEE

Joel Dendy
Los Alamos National Laboratory

Craig Douglas
IBM/Yale University

Paul Frederickson
RIACS

Van Henson
Naval Postgraduate School

Jan Mandel
University of Colorado at Denver

Tom Manteuffel
University of Colorado

Steve McCormick
University of Colorado

Duane Melson
NASA Langley Research Center

Seymour Parter
University of Wisconsin

Joseph Pasciak
Brookhaven National Laboratory

John Ruge
University of Colorado at Denver

Klaus Stueben
Gesellschaft f. Math. u. Datenverarbeitung

Pieter Wesseling
Delft University

Olof Widlund
Courant Institute

ATTENDEES

LOYCE ADAMS	adams@amath.washington.edu
FERNANDO ALVARADO	alvarado@engr.wisc.edu
EYAL ARIAN	arian@icase.edu
STEVE ASHBY	sfashby@llnl.gov
VICTOR BANDY	vab@swan.lanl.gov
DANA BEDIVAN	bedivan@utamat.uat.edu
M. BERNDT	berndt@colorado.edu
PAVEL BOCHEV	bochev@utamat.uta.edu
JAMES BORDNER	bordner@cs.uiuc.edu
A. BORZI	Alfio.Borzi@comlab.ox.ac.uk
ACHI BRANDT	mabrandt@weizmann.weizmann.ac.il
SUSANNE BRENNER	brenner@math.scarolina.edu
MARIAN BREZINA	mbrezina@tiger.cudenver.edu
OLIVER BROKER	broker@cs.colorado.edu
JAN BROEZE	j.broeze@math.utwente.nl
ZHIQIANG CAI	zcai%mathd.usc.edu@usc.edu
XIAO-CHUAN CAI	cai@schwarz.cs.colorado.edu
PHIL CALVIN	mudpuppy@gibbs.oit.unc.edu
DAVID CANRIGHT	dcanright@nps.navy.mil
MARIO CASARIN	casarin@math1.nyu.edu
RICHARD CASEY	richard.casey@asu.edu
ZHANGXIN CHEN	zchen@golem.math.smu.edu
REGINALD W. CLEMENS	reg@dwf.com (or) clemens@plk.af.mil
A. W. CRAIG	Alan.Craig@sima.sintef.no
GENE D'YAKANOV	dknv@cmc.msk.su
BRUCE DAVIS	davis@cfdlab.ae.utexas.edu
JOEL DENDY	jed@lanl.gov
QINGPING DENG	deng@math.utk.edu
CRAIG DOUGLAS	douglas-craig@cs.yale.edu
JON DYM	jdym@cams.usc.edu

HOWARD ELMAN	elman@cs.umd.edu
BJORN ENGQUIST	engquist@math.ucla.edu
R. D. FALGOUT	falgout@bacchus.llnl.gov
CHARBEL FARHAT	charbel@alexandra.colorado.edu
HERMANN FASEL	faselh@ccit.arizona.edu
JEAN MICHEL FIARD	fiard@newton.colorado.edu
PAUL FREDERICKSON	MathCube@aol.com
KLAUS GARTNER	gaertner@iis.ee.ethz.ch
THOR GJESDAL	thor@cmr.no
SIMON GLEYZER	gleyzer@gibbs.oit.unc.edu
WOJCIECH GOLIK	golik@arch.umsl.edu
HERVE GUILLARD	Herve.Guillard@inria.fr
VAN HENSON	vhenson@boris.math.nps.navy.mil
ALAN HEROD	aherod@newton.colorado.edu
GREGORY HILL	ghill@cs.colorado.edu
LOUIS HOWELL	nazgul@bigbird.llnl.gov
JEROME JAFFRE	Jerome.Jaffre@inria.fr
JIM JONES	jijones@mtha.usc.edu
KIRK JORDAN	kjordan@vnet.ibm.com
MICHAEL JUNG	Dr.Michael.Jung@mathematik.tu-chemnitz
DAVID KINCAID	kincaid@cs.utexas.edu
AXEL KLAWONN	klawonn@goedel.uni-muenster.de
ANDREW KNYAZEV	aknyazev@tiger.cudenver.edu
HWAR-CHING KU	ku@aplcomm.jhuapl.edu
CHEN-YAO G. LAI	cylai@math.ccu.edu.tw
R. LAZAROV	lazarov@math.tamu.edu
CHANG OCK LEE	colee@math.inha.ac.kr
BARRY LEE	blee@boulder.colorado.edu
G. SCOTT LETT	slett@ssii.com
YONG LI	lyong@digger.gsfc.nasa.gov
C. LIU	cliu@carbon.denver.colorado.edu
ZHINING LIU	zliu@evans.denver.colorado.edu

SERGUEI MALIASSOV	malyasov@isc.tamu.edu
JAN MANDEL	jmandel@carbon.denver.colorado.edu
TOM MANTEUFFEL	tmanteuf@newton.colorado.edu
TAREK MATHEW	mathew@ledaig.uwyo.edu
STEVE MCCORMICK	stevem@newton.colorado.edu
S. MCKAY	mckay@math.byu.edu
ROBERT MCLAY	mclay@cfdlab.ae.utexas.edu
A .J. MEIR	ajm@math.auburn.edu
DUANE MELSON	n.d.melson@larc.nasa.gov
ILYA MISHEV	mishev@isc.tamu.edu
WILLIAM MITCHELL	mitchell@cam.nist.gov
HANS MOLENAAR	hansmo@twi.tudelft.nl
SERGEI NEPOMNYASCHIKH	svnep@comcen.nsk.su
JOHN W. NEUBERGER	jwn@unt.edu
ELYAS NURGAT	nurgat@scs.leeds.ac.uk
SUELY OLIVEIRA	suely@cs.tamu.edu
MARY OMAN	imsgmoma@math.montana.edu
MARIA ELIZABETH ONG	ong@sdna5.ucsd.edu
ROSSEN PARASHKEVOV	rossen@newton.colorado.edu
SEYMOUR PARTER	parter@cs.wisc.edu
JOE PASCIAK	pasciak@bnl.gov
JAN PEETERSWEEM	peetersw@newton.colorado.edu
CHRISTOPH PFLAUM	pflaum@informatik.tu-muenchen.de
J. R. PHILLIPS	jphill@rle-vlsi.mit.edu
KLAUS RESSEL	kressel@tiger.cudenver.edu
KRIS RIEMSLAGH	Kris.Riemslagh@rug.ac.be
GUY ROBINSON	robinson@npac.syr.edu
JOHN RUGE	jruge@boulder.colorado.edu
TORGEIR RUSTEN	Torgeir.Rusten@si.sintef.no
FAISAL SAIED	saied@cs.uiuc.edu
MARKUS SARKIS	msarkis@tigger.cs.colorado.edu
AIHUA SHAKER	ashaker@afit.af.mil

YAIR SHAPIRA	yair@csc.cs.technion.ac.il
DAVID SIDILKOVER	sidilkov@icase.edu
PETER STAAB	staab@newton.colorado.edu
GERHARD STARKE	starke@boulder.colorado.edu
ANDREAS STATHOPOULOS	andreas@vuse.vanderbilt.edu
WILLIAM J. STEWART	
DANIEL B. SZYLD	szyld@euclid.math.temple.edu
RADEK TEZAUR	tezaur@tiger.cudenver.edu
MARIETTA TRETTER	eo21mt@tamvm1.tamu.edu
ALEXANDER TROFIMOV	fmm@uni.tiv.dnepropetrovsk.ua
STEFAN VANDEWALLE	stefan@ama.caltech.edu
PETR VANEK	pvanek@tiger.cudenver.edu
PRATAP VANKA	vanka@uy.ncsa.uiuc.edu
APOSTOL VASSILEV	apostol@isc.tamu.edu
C. VUIK	cvuik@math.tudelft.nl
HONG WANG	hwang@math.scarolina.edu
JUNPING WANG	junping@schwarz.uwyo.edu
RUIKE WANG	wang@rsci.ssii.com
OLOF WIDLUND	widlund@widlund.cs.nyu.edu
STEPHEN B. WINEBERG	wineberg@math.lsa.umich.edu
KRISTIAN WITSCH	witsch@numerik.uni-duesseldorf.de
DEXUAN XIE	xie@math.uh.edu
JINCHAO XU	xu@math.psu.edu
IRAD YAVNEH	irad@cs.technion.ac.il
DAVID YOUNG	young@cs.utexas.edu
XIUYANG YU	xyu@carbon.denver.colorado.edu
LEONID ZASLAVSKY	zasl@wisdom.weizmann.ac.il
X. ZHENG	xzheng@tiger.cudenver.edu

MULTIGRID HISTORY

(At the awards ceremony of the conference, Achi Brandt presented the following history of multigrid. The reader should study the truths contained herein and revel in the humor.)

The early history of multigrid has recently become a hot subject of research. An ancient multigrid code was uncovered during extensive excavations last year in northern Turkestan. Carbon tests indicate that this code has an efficiency of 5.1 on the Richter scale. Some researchers believe that the V cycle was practiced by the Neanderthals. The use of the Full Multigrid (FMG) algorithm was, however, unique to Homo sapiens and is one of the major reasons for their ultimate survival. Prototypes of two-grid algorithms predate the first hominids. Most historians agree that coarsening was, in fact, invented by the dinosaurs; however, coarse-to-fine grid transfers were unknown to them, which explains their extinction.

Earlier geological findings include rich multilevel deposits that have been unearthed in several North American gold mines, and thick layers of old multigridders have been discovered at Copper Mountain.

The artifacts at the northern Turkestan site indicate that an early form of residual weighting was already in widespread use before the middle Full Approximation Storage (FAS) period. When Copernicus first introduced line relaxation, it was banned by the Catholic church. Pope Pointus the Square decreed that mere mortals should not practice such nonlocal schemes. He feared this practice would lead humanity to incompleteness, in particular to the incomplete LU decomposition of the Dutch church. The advent of variational coarsening during the French Revolution marks the dawn of the modern era, which is quite familiar to us all.

Page intentionally left blank

CONTENTS

Part 2*

*Part 2 is presented under separate cover.

Page intentionally left blank

A MULTIGRID ALGORITHM FOR IMMERSED INTERFACE PROBLEMS

Loyce Adams [1]
Dept. of Applied Mathematics
University of Washington

SUMMARY

Many physical problems involve interior interfaces across which the coefficients in the problem, the solution, its derivatives, the flux, or the source term may have jumps. These interior interfaces may or may not align with a underlying Cartesian grid. Zhilin Li, in his dissertation, showed how to discretize such elliptic problems using only a Cartesian grid and the known jump conditions to second order accuracy. In this paper, we describe how to apply the full multigrid algorithm in this context. In particular, the restriction, interpolation, and coarse grid problem will be described. Numerical results for several model problems are given to demonstrate that good rates can be obtained even when jumps in the coefficients are large and do not align with the grid.

1. INTRODUCTION

Many physical problems involve interior interfaces across which the coefficients in the problem, the solution, its derivatives, the flux, or the source term may have jumps. These interior interfaces may or may not align with a underlying Cartesian grid. As an example, single phase Darcy flow in porous media is governed by the equation $\nabla \cdot (\beta \nabla p) = 0$ for the pressure p where $\beta = \kappa/\mu$ with κ the permeability and μ the viscosity. If the medium has an interface across which the permeability varies, we know that $[p] = 0$ and $[\beta p_n] = 0$ at this interface. Another example is Stokes flow where the interface is the boundary of a moving membrane or bubble, ([1], [2]). A more complicated problem is to model the blood flow in the human heart. Here the interface is the boundary of the heart. Peskin [3] solves for the velocity of the fluid in which the heart is immersed by solving the Navier-Stokes equations on a Cartesian grid with a delta function forcing term determined by the force the heart wall exerts on the fluid. It can be shown [3] that this singular source term in the Navier-Stokes equations leads to jumps in pressure and the derivatives of velocity across the interface, and is discretized by discrete delta functions and transfered to the nearby Cartesian grid points. The velocity of the fluid is then used to move the boundary of the heart to the next time. This procedure is called the *immersed boundary method* and seems to be only first order accurate due to the way the force on the interface is spread to the Cartesian grid.

Zhilin Li has recently developed an approach for discretizing elliptic problems with interior interfaces called the *immersed interface method* (IIM), ([4], [5]), which can handle both discontinuous coefficients and singular sources. The idea is to compute on a Cartesian grid only, as in Peskin's

[1] This work was supported in part by the Scientific Computing Division of the National Center for Atmospheric Research, which is supported by NSF, and in part by Department of Energy grant DE-FG06-93ER25181 and NSF grant DMS-9303404.

immersed boundary method, but to find accurate discretization stencils by incorporating knowledge of where the interface is located and the known jumps in the solution there, rather than by smearing the force with a discrete delta function. Li showed that second-order accurate discretizations could be found for a wide class of problems. Of course, there are problems of physical interest where the jumps at the interface are not known a priori and must be solved for first before such an approach can be taken. Such problems and solution techniques are discussed in [6], but will not be the focus of this paper.

The purpose of this paper is to describe how the full multigrid method (FMG) can be applied to the discrete equations that result from the IIM. Many authors have given efficient multigrid schemes for both symmetric and nonsymmetric systems of equations that arise from elliptic problems with discontinuous coefficients. A partial list includes [7], [8], [9], [10], [11], [12], [13], and [14]. For problems with discontinuous coefficients, care must be taken to devise a proper method of interpolation for the multigrid process. Much of the work done in this direction has assumed that any interfaces are aligned with the grid. However, Aaron Fogelson and James Keener have used multigrid schemes to solve non-aligned immersed interface problems for two-dimensional heat equations in regions with holes, and to solve for electrical potentials in cardiac tissues, [15].

One common approach is to use what is called operator-induced interpolation. That is, the stencil for the partial differential equation incorporates information about the jumps in the coefficients, and this stencil can be modified to produce a stencil for interpolation. Such an idea is found in [8] and [9] and has the advantage that explicit information about the interface need not be known directly. Black box multigrid can find out from the problem stencil how to interpolate. This approach presumably can be used when the interface does not align with the grid, assuming the problem was discretized accurately. In the future, we plan to try this approach in conjunction with an IIM discretization.

Here, we present a different approach. Since our stencil for the problem comes from the IIM, we have all the information about the interface and the jumps there. In this paper, we show how to use this information to build an $O(h^2)$ accurate interpolation scheme. The results of this approach seem promising since V-cycle rates of .06 to .13 have been achieved.

The paper is organized as follows. Section 2 gives an overview of the IIM. Section 3 describes our multigrid scheme with a derivation of the modified bilinear interpolation. Section 4 gives numerical results. Section 5 states the conclusions and avenues for further work.

2. IMMERSED INTERFACE METHOD OVERVIEW

In this section, we review the immersed interface method. Details can be found in [4] and [5]. The IIM provides a discretization of elliptic PDEs that is $O(h^2)$, where h is the uniform mesh spacing in both the x and y directions. Consider the problem

$$(1) \qquad \begin{aligned} (\beta u_x)_x + (\beta u_y)_y &= f(x,y) \text{ in } \Omega \\ [u]_\Gamma &= w(s) \end{aligned}$$

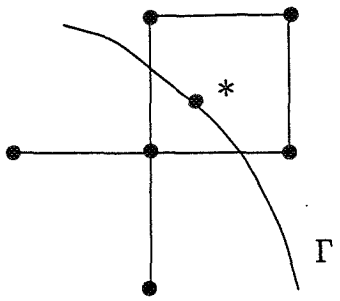


FIG. 1. *6-pt Stencil for Irregular Points*

$$[\beta u_n]_\Gamma \;=\; v(s)$$

where boundary conditions on Ω are given and Γ is the interface, across which the jump in the solution and flux are assumed known as functions of the arc length s. The stencil for a regular point (all points of the standard 5-point stencil are on the same side of the interface) is the usual $O(h^2)$ approximation that uses the 5-point stencil for u values and its edge midpoints for β values. To discretize (1) at an irregular point, Li uses a sixth point stencil as shown in Figure 1 where * represents a point (x^*, y^*) on the immersed interface and looks for a formula at the center point of the form

$$(\beta u_x)_x + (\beta u_y)_y = \sum_1^6 \gamma_i u_i - c + O(h) \tag{2}$$

where u_i denotes the i-th point in the 6-point stencil, the γ_i's are the coefficients to be determined, and c is a correction term that can be computed once the γ_i's are known. Requiring the truncation error in (2) to be $O(h)$ at the irregular points and the truncation error to be $O(h^2)$ at the regular points is sufficient to guarantee that a global error of $O(h^2)$ is achieved everywhere.

Let ξ and η be the normal and tangential directions at the point (x^*, y^*) which are given by

$$\begin{aligned} \xi &= (x - x^*)\cos\theta + (y - y^*)\sin\theta \\ \eta &= -(x - x^*)\sin\theta + (y - y^*)\cos\theta. \end{aligned} \tag{3}$$

We then expand u_i about the point (x^*, y^*) on the interface after changing to the (ξ, η) variables. That is,

$$u_i = u^* + \xi_i u^*_\xi + \eta_i u^*_\eta + \xi_i \eta_i u^*_{\xi\eta} + \frac{1}{2}\xi_i^2 u^*_{\xi\xi} + \frac{1}{2}\eta_i^2 u^*_{\eta\eta} + \ldots \tag{4}$$

where * means to take the + or - limiting value on the outside or inside of the interface, respectively. Then we have 12 unknown terms on the right hand side in (2) and 6 unknown γ_i's. But, since we know the jumps from (1), the following jump conditions can be derived for the special case where $\beta = \beta_{in}$ inside the interface and $\beta = \beta_{out}$ outside.

1. $u^+ = u^- + w$
2. $u^+_\eta = u^-_\eta + w'$
3. $u^+_\xi = \rho u^-_\xi + v/\beta_{out}$
4. $u^+_{\xi\eta} = \rho u^-_{\xi\eta} + (1-\rho)\Psi'' u^-_\eta + w'\Psi'' + v'/\beta_{out}$
5. $u^+_{\eta\eta} = u^-_{\eta\eta} + (1-\rho)\Psi'' u^-_\xi + w'' - \Psi'' v/\beta_{out}$
6. $u^+_{\xi\xi} = \rho u^-_{\xi\xi} + (\rho-1)\Psi'' u^-_\xi + (\rho-1)u^-_{\eta\eta} + \Psi'' v/\beta_{out} - w'' + [f]/\beta_{out}$

The variables w and v are functions of η only, $\rho = \beta_{in}/\beta_{out}$, the interface is described parametrically as $\xi = \Psi(\eta)$, and all variables in the conditions above have been evaluated at $(\xi, \eta) = (0, 0)$ which corresponds to the * point on the interface. Next, we substitute these six conditions into (4) and then substitute (4) into (2) to get six equations in six unknowns for the γ_i's. Once these are found, c is determined from the γ_i's and the jump conditions. The end result gives an $O(h^2)$ approximation to the exact solution u that satisfies $\beta(u_{xx} + u_{yy}) = f$.

To use the IMM to generate the problem, the user must specify w, v, and $[f]$ at control points (X, Y) along the interface. The program fits a cubic spline through X, Y, v, w, and $[f]$ at these control points to define $X(s)$, $Y(s)$, $v(s)$, $w(s)$, and $[f](s)$ as functions of the arc length parameter s. The quantitites in the jump relations are then derivable from these functions. As part of the procedure, each grid point is typed as being inside, outside, or on the interface, as well as being regular or irregular.

One advantage of this approach is that the same interface can be used on each grid of a multigrid routine. That is, as we refine the grid, we need not refine a grid representing the interface. It is sufficient to specify a relatively small number of control points, depending on the smoothness of the interface, in order to describe the interface with a spline. Of course, this procedure can not handle problems with interfaces that can not be well represented with a cubic spline. A future improvement to the implementation of the IIM would be to describe the immersed interface with a level set formulation. The coding involved would be reduced significantly and we plan to do this before we tackle problems with multiple interfaces.

3. A FULL MULTIGRID SCHEME

The result of the IIM is a discrete system of equations, $A^h u^h = f^h$, on the finest grid with uniform mesh spacing h in each coordinate direction. The goal is to develop a multigrid strategy to solve this system quickly. Unlike the Black Box multigrid approach of [8] and [9] which uses operator-induced interpolation, we base our strategy on knowledge of where the interface is located and the jumps there. We have not yet compared our approach to Dendy's but we can claim to get fairly good multigrid rates with our approach for this class of problems.

The basic components of full multigrid are the smoother, the restriction operator, the interpolation operator, and the coarse grid problem. We now describe what we choose for each. For all our test cases, point-rowwise Gauss-Seidel worked fine as the smoother. More complicated problems with larger jumps in the coefficients may require a more sophsicated smoother. The coarse grid problem, $A^{2h}u^{2h} = f^{2h}$ was taken to be the output of the IIM method with mesh size $2h$. This

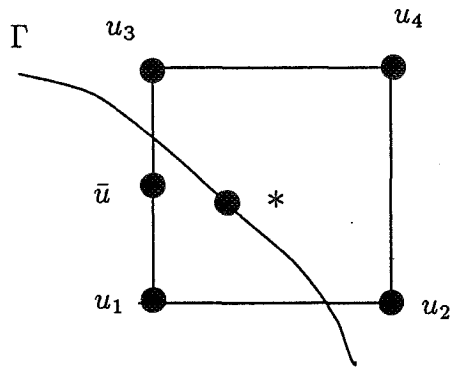


FIG. 2. *Interpolation for* $\bar{u}$

choice seems to limit the size of the coarsest grid to be 10×10 ($h = .4$) for problems with ratios $\beta_{in}/\beta_{out} = 2000$. It is possible to define A^{2h} to satisfy the Galerkin condition, $A^{2h} = I_h^{2h} A^h I_{2h}^h$, but this has not been implemented yet. With the exception of the limitation in grid size described above, our definition of the coarse grid problem worked fine.

The interpolation operator we used is a modified bilinear interpolation in the ξ, η coordinates for grid cells that contain an interface. If the cell does not contain an interface, the interpolation reduces to ordinary bilinear interpolation. To interpolate to the fine grid point at the center of a coarse grid cell, we build a formula based on the corner values of this cell plus a correction term. To interpolate to the midpoint of a vertical(horizontal) edge we choose either the cell to the east or west (or north or south) and find a formula based on the corner values of this chosen cell plus a correction term. The cell choice depends on the location of the interface relative to the fine grid point for which we are seeking an interpolated value. For example, if one cell is regular (no interfaces crossing its boundary) it is preferred over the irregular cell. If both cells are irregular, an attempt is made to choose the one that will produce the most accurate value.

To describe the scheme mathematically, we consider the chosen coarse grid cell shown in Figure 2 where u_i are the four coarse grid values, $\bar{u}$ is a fine grid point whose value we wish to find, and $*$ is a point (x^*, y^*) on an interface cutting through the cell. During the continuation phase of FMG, we look for a solution to $\bar{u}$ of the form

$$\bar{u} = \sum_1^4 \gamma_i u_i - c \tag{5}$$

much in the same way as the 6-point stencil was found for the PDE in the IIM method. Again, let ξ_i and η_i be the transformed variables given in (3) of the previous section, and expand each u_i and $\bar{u}$ about (x^*, y^*) on the interface using (4). Using the jump conditions given for the IIM method, we get the system $A_\gamma \gamma = b_\gamma$ for the γ_i's after equating the coefficients of u^-, u_ξ^-, u_η^-, and $u_{\xi\eta}^-$. The matrices A_γ and b_γ are given below,

$$A_\gamma = \begin{bmatrix} 1 & 1 & 1 & 1 \\ \xi_1\rho_1 & \xi_2\rho_2 & \xi_3\rho_3 & \xi_4\rho_4 \\ \eta_1 + \alpha_1\xi_1\eta_1\tau_1 & \eta_2 + \alpha_2\xi_2\eta_2\tau_2 & \eta_3 + \alpha_3\xi_3\eta_3\tau_3 & \eta_4 + \alpha_4\xi_4\eta_4\tau_4 \\ \xi_1\eta_1\rho_1 & \xi_2\eta_2\rho_2 & \xi_3\eta_3\rho_3 & \xi_4\eta_4\rho_3 \end{bmatrix} \tag{6}$$

$$b_\gamma = \begin{bmatrix} 1 \\ \bar{\xi}\bar{\rho} \\ \bar{\eta} + \bar{\alpha}\bar{\xi}\bar{\eta}\bar{\tau} \\ \bar{\xi}\bar{\eta}\bar{\rho} \end{bmatrix} \tag{7}$$

and the correction term $c = c_1 - \bar{c}$ where

$$\begin{aligned} c_1 &= \sum_1^4 \alpha_i\gamma_i(w + \xi_i v/\beta_{out} + \eta_i w' + \xi_i\eta_i(w'\Psi'' + v'/\beta_{out})) \\ \bar{c} &= \bar{\alpha}(w + \bar{\xi}v/\beta_{out} + \bar{\eta}w' + \bar{\xi}\bar{\eta}(w'\Psi'' + v'/\beta_{out})). \end{aligned} \tag{8}$$

In the above equations, if the cell is regular, $\alpha_i = 0$ and $\bar{\alpha} = 0$. If the cell is irregular, $\alpha_i = 1$ if the point (x_i, y_i) is outside the interface and $\alpha_i = 0$ if it is inside or on the interface. Likewise, if the cell is irregular, $\bar{\alpha} = 1$ if the point $(\bar{x}, \bar{y})$ is outside the interface and $\bar{\alpha} = 0$ if it is inside or on the interface. If the point (x_i, y_i) is outside the interface then $\rho_i = \beta_{in}/\beta_{out}$ and $\rho_i = 1$ if it is inside or on the interface. Likewise, $\bar{\rho} = \beta_{in}/\beta_{out}$ if the point $(\bar{x}, \bar{y})$ is outside the interface and $\bar{\rho} = 1$ if it is inside or on the interface. Also $\tau_i = (1 - \rho_i)\Psi''$, and $\bar{\tau} = (1 - \bar{\rho})\Psi''$.

Upon examination of these equations, it can be seen that for each irregular coarse grid cell, we really are calculating two bilinear functions, each of the form $\bar{u} = a + b\xi + c\eta + d\xi\eta$. Each function interpolates the coarse grid points on the respective side of the interface (4 conditions). In addition, the functions are such that the jump conditions $[u]_\Gamma$, $[\beta u_\xi]_\Gamma$, $[u_\eta]_\Gamma$, and $[\beta u_{\xi\eta}]_\Gamma$ are satisfied at the interface point (x^*, y^*) (the remaining four conditions). Since the terms left off are $O(h^2)$, the formula is $O(h^2)$ for $\bar{u}$, relative to the true solution of the partial differential equation. Hence, these formulas should give good results if the second derivatives are not too large relative to the mesh spacing.

During the V-cycle, we need to interpolate the error e^{2h} to the finer grid. The same approach could be used if we knew $[e^{2h}]_\Gamma$ and $[\beta e_n^{2h}]_\Gamma$ at the interface. These are not known, but if the smoother is doing a good job, it makes sense to set these jumps to zero. Then the same γ_i's that were calculated during the continuation phase for interpolating u are the proper values to use for interpolating the error as well. This approach works well in practice as seen in the results in the next section.

We choose the restriction operator to be a multiple of the transpose of the interpolation operator just described. In particular, $I_h^{2h} = .25(I_{2h}^h)^T$. In the case of regular cells, this reduces to full

weighting. For irregular cells, the stencil has a width of two grid cells in each direction, excluding other coarse grid points. The data structure used is a 5×5 stencil with other coarse grid connections set to zero.

4. NUMERICAL RESULTS

Several test problems were run using the full multigrid scheme described above. For each test problem, we use the notation V(a,b) to denote that a pre- and b post- smooths were used in each V- cycle. More cycles than necessary to reach truncation error were taken for the purpose of studying the convergence to the solution of the discrete system. In all problems, about 3 V-cycles were sufficient to reach truncation level. In each Table, *derr* denotes the difference between the computed solution and the exact solution of the difference equations and *res* is the residual. The grid size given for each Table is that of the finest grid. In the Figures, *err* is the difference between the computed solution and the exact solution of the partial differential equation.

Problem 1

The domain Ω is the $(-2,2) \times (-2,2)$ square and the interface Γ is the unit circle. The problem is

$$\beta(u_{xx} + u_{yy}) = f$$
$$\beta_{in} = .5,\ \beta_{out} = 1000.,\ f_{in} = 2.0,\ f_{out} = 0$$
$$u_{in} = x^2 + y^2,\ u_{out} = x^2 - y^2$$
$$[u]_\Gamma = -2y_1^2$$
$$[\beta u_n]_\Gamma = 2\beta_{out}(x_1^2 - y_1^2) - 2\beta_{in}$$

Table 1 shows rates of each V-cycle to be .13 for both the discrete error and the residual for a 2-level scheme on a 40×40 grid with 2-pre and 2-post smooths. Notice that the modified bilinear interpolation used in continuation gave a starting guess on the finest grid of .02. This is good since the mesh size on the finest grid is $h = .1$.

Cycle	$\|\|derr\|\|_\infty$	$\|\|res\|\|_\infty$	$rate_{derr}$	$rate_{res}$
Starting	$.20 \times 10^{-1}$	$.50 \times 10^{4}$		
1-V	$.23 \times 10^{-2}$	$.13 \times 10^{3}$	.12	.03
2-V	$.26 \times 10^{-3}$	$.57 \times 10^{1}$	.11	.05
3-V	$.34 \times 10^{-4}$	$.76 \times 10^{0}$	.13	.13
4-V	$.44 \times 10^{-5}$	$.96 \times 10^{-1}$	.13	.13
5-V	$.56 \times 10^{-6}$	$.12 \times 10^{-1}$	.13	.13
6-V	$.73 \times 10^{-7}$	$.16 \times 10^{-2}$	.13	.13
7-V	$.94 \times 10^{-8}$	$.21 \times 10^{-3}$	.13	.13

Table 1. Problem 1: V(2,2), 40x40, 2-levels

Table 2 gives 2-level V(4,4) results for Problem 1. Notice that the rates went down from .13 to .06.

Cycle	$\|\|derr\|\|_\infty$	$\|\|res\|\|_\infty$	$rate_{derr}$	$rate_{res}$
Starting	$.20 \times 10^{-1}$	$.50 \times 10^{4}$		
1-V	$.37 \times 10^{-3}$	$.91 \times 10^{1}$	.02	.002
2-V	$.25 \times 10^{-4}$	$.34 \times 10^{0}$	.07	.037
3-V	$.14 \times 10^{-5}$	$.20 \times 10^{-1}$	.06	.06
4-V	$.93 \times 10^{-7}$	$.13 \times 10^{-2}$	.07	.07
5-V	$.60 \times 10^{-8}$	$.83 \times 10^{-4}$	.06	.06
6-V	$.39 \times 10^{-9}$	$.53 \times 10^{-5}$	.06	.06
7-V	$.25 \times 10^{-10}$	$.34 \times 10^{-6}$	.06	.06

Table 2. Problem 1: V(4,4), 40x40, 2-levels

Table 3 gives 3-level results for an 80×80 fine grid and V(4,4). Notice that we still get rates of .06 with 3-levels. Also note that even though the level 2 problem was solved with only 1 V-cycle, the starting error for level 3 was .016.

Cycle	$\|\|derr\|\|_\infty$	$\|\|res\|\|_\infty$	$rate_{derr}$	$rate_{res}$
Starting	$.16 \times 10^{-1}$	$.15 \times 10^{5}$		
1-V	$.96 \times 10^{-3}$	$.28 \times 10^{3}$	.06	.02
2-V	$.15 \times 10^{-4}$	$.28 \times 10^{1}$	.02	.01
3-V	$.78 \times 10^{-6}$	$.64 \times 10^{-1}$	.05	.02
4-V	$.51 \times 10^{-7}$	$.38 \times 10^{-2}$	.06	.06
5-V	$.28 \times 10^{-8}$	$.14 \times 10^{-3}$	.05	.04
6-V	$.19 \times 10^{-9}$	$.90 \times 10^{-5}$	.07	.06

Table 3. Problem 1: V(4,4), 80x80, 3-levels

Figure 3 shows the computed solution for this problem with V(4,4) for an 80×80 fine grid after 7 V-cycles. Notice the sharpness of the jump at the interface. Figure 4 shows the associated *err*. This error, $O(10^{-5})$, is concentrated along the interface as expected since the truncation error is largest there. We note that the discrete error, *derr*, is $O(10^{-11})$ and is much smoother at the interface due to the multigrid smoothing.

PROBLEM 2

The problem domain Ω is the $(-2,2) \times (-2,2)$ square and the interface Γ is the unit circle. The problem is

$$\beta(u_{xx} + u_{yy}) = f$$

$$\beta_{in} = 1,\ \beta_{out} = 1000,\ f_{in} = f_{out} = 2000$$

$$u_{in} = 1000x^2,\ u_{out} = x^2$$

$$[u]_\Gamma = -999(x_1^2)$$

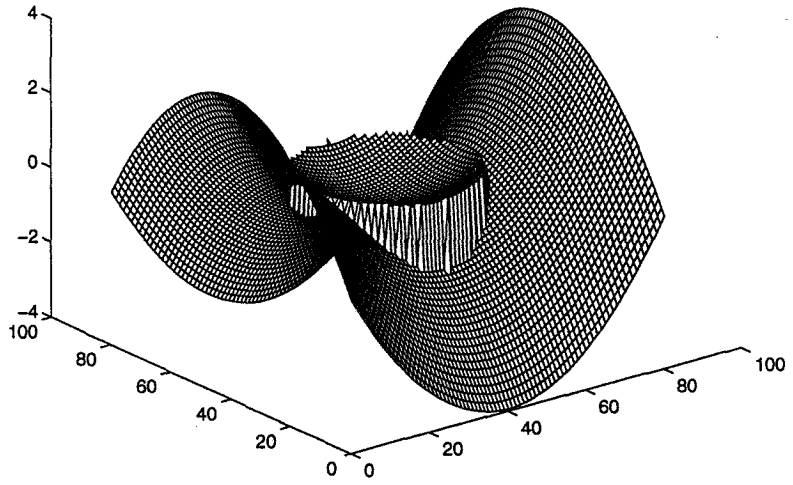


FIG. 3. *u for Problem 1.*

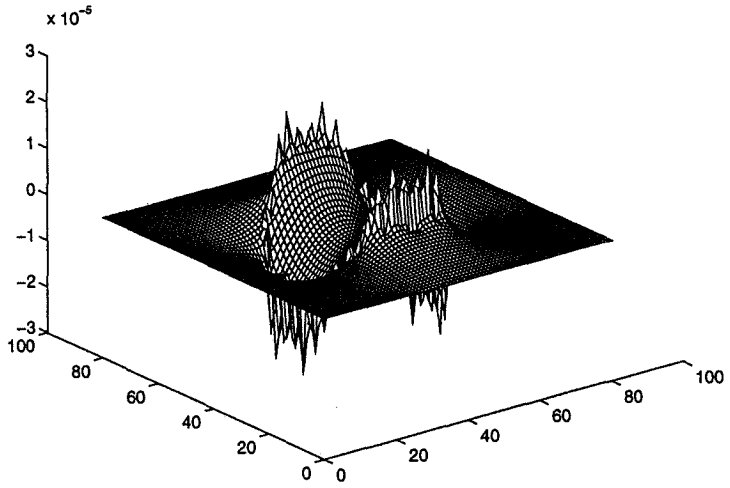


FIG. 4. *err for Problem 1.*

$$[\beta u_n]_\Gamma = 0,\ [u_n]_\Gamma = -999(2x_1^2)$$

Note that this problem has a jump in the normal derivative at the interface even though the jump in the flux is zero. Table 4 shows rates for a 2-level method with V(4,4) to be .03 for the discrete error and .06 for the residual.

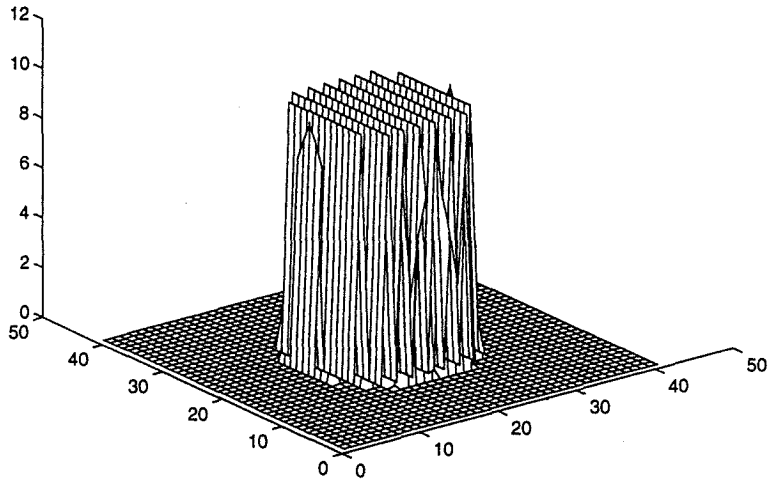


FIG. 5. *Starting err for Problem 2.*

Cycle	$\|\|derr\|\|_\infty$	$\|\|res\|\|_\infty$	$rate_{derr}$	$rate_{res}$
Starting	$.10 \times 10^{2}$	$.20 \times 10^{5}$		
1-V	$.19 \times 10^{0}$	$.25 \times 10^{2}$	.02	.001
2-V	$.48 \times 10^{-2}$	$.59 \times 10^{0}$	.03	.02
3-V	$.12 \times 10^{-3}$	$.12 \times 10^{-2}$	.03	.02
4-V	$.36 \times 10^{-5}$	$.74 \times 10^{-3}$	.03	.06
5-V	$.11 \times 10^{-6}$	$.49 \times 10^{-4}$	.03	.06
6-V	$.30 \times 10^{-8}$	$.31 \times 10^{-5}$	.03	.06
7-V	$.11 \times 10^{-9}$	$.20 \times 10^{-6}$	.04	.06

Table 4. Problem 2: V(4,4), 40x40, 2-levels

Of special note in Table 4 is the starting error produced by the modified bilinear interpolation during continuation. At first sight this error of 10 looks quite bad. But notice that $u_{xx} = 2000$ for points inside the interface, and the term $\frac{1}{2}(x - x^*)^2 u_{xx}$ that is not included in the bilinear interpolation is exactly 10. In fact, Figure 5 shows the starting error to be very sharp at the interface, reflecting the fact that the truncation error has a different constant for points inside and outside the interface. This is the best we can hope to accomplish with bilinear interpolation for this problem. We do not plot the solution and error for this problem since the graphs are quite similar to Problem 1 in that the jumps are captured very sharply.

Problem 3

As an application we consider single-phase saturated flow governed by Darcy's law,

$$\begin{aligned} \vec{u} &= -\beta \nabla p \\ \nabla \cdot \vec{u} &= 0 \end{aligned} \tag{9}$$

where $\vec{u} = (u, v)^T$ is the velocity vector and $\beta = \kappa/\mu$ with a discontinuous permeability at the interface. Such problems arise in groundwater flow and contaminant transport. Combining the equations above we get the elliptic equation

$$\begin{aligned} \nabla \cdot (\beta \nabla p) &= 0 \\ [p]_\Gamma &= 0 \\ [\beta p_n]_\Gamma &= 0 \end{aligned} \tag{10}$$

for the pressure p. Equation (10) is then discretized with the IIM and solved using multigrid. The velocities of the flow are then determined from (9).

A similar strategy that was used for modified bilinear interpolation can be used to devise an $O(h)$ formula for p_x and p_y in cells with interfaces. One could also get $O(h)$ formulas by using one-sided differences on the correct side of the interface. If the pressures, p, are calculated by multigrid on a grid of size h, modified bilinear interpolation can be used to give p at cell-centers and edges on a grid of size $h/2$. Then the needed information is available to find derivatives to $O(h)$ at grid points of the $h/2$ grid. A more exact, though more expensive method, is to calculate pressures on a grid of size $h/2$ for use in the derivative calculation on a grid of size h. This was the approach that was taken in the results that follow.

Once derivatives are found, we solve the equation

$$q_t + \vec{u} \cdot \nabla q = 0 \tag{11}$$

for advection of a contaminant with concentration q. This is done with LeVeque's **Clawpack** software on a uniform grid, ([16], [17]). For the test problem we take Ω to be the $(-2, 2) \times (-2, 2)$ square and Γ to be the interface shown in Figure 6. On the square, $p = 1$ at the left boundary, $p = 0$ at the right boundary, and $p_y = 0$ at the top and bottom boundaries. The permeabilities are $\beta = 5$ inside the interface and $\beta = 1$ outside the interface. Initially, $q = 0$ and at the left(inflow) boundary $q = 1$, and an 80×80 computational grid is used.

Figure 6 shows the velocities that were determined by differencing the pressure that came out of the multigrid routine. Since β is larger inside the interface, the velocity should move the

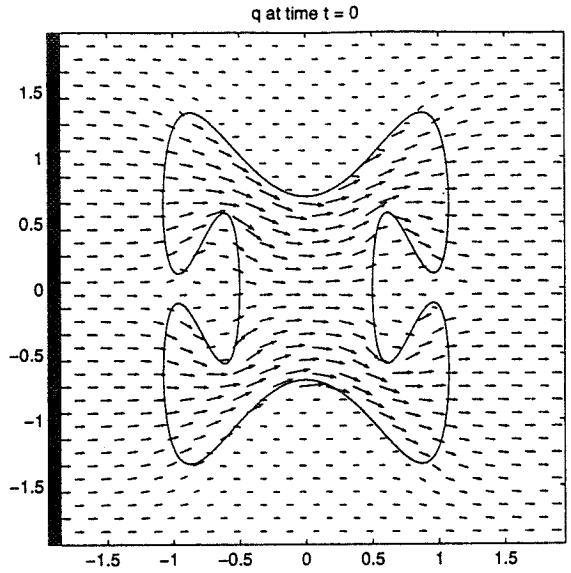


FIG. 6. *Velocities for Problem 3.*

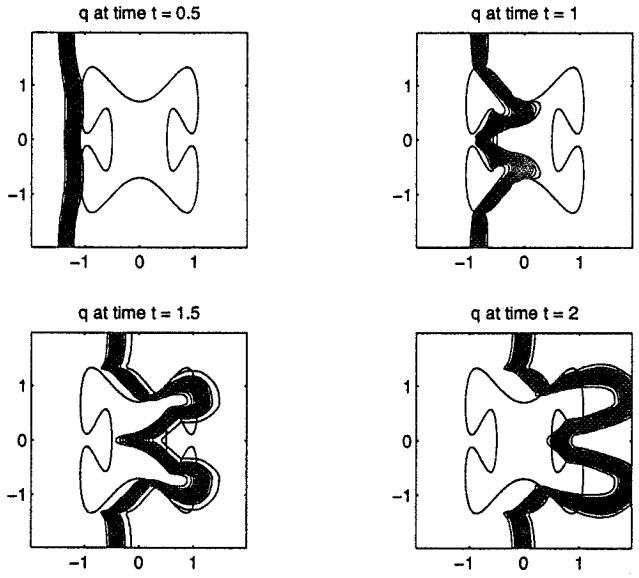


FIG. 7. *Contours of q for Problem 3.*

contaminant quicker through this region than around it. This is what is observed at four times as shown in Figure 7. Our approach did give sharp results for the moving front of the contaminant even though the **Clawpack** routine used did not have knowledge of where the interface was located.

5. CONCLUSIONS

We have demonstrated that a full multigrid algorithm can be designed for interface problems where the jumps in coefficients, solution, derivatives, flux, or source term are not aligned with the underlying Cartesian grid. This algorithm correctly solves the fine grid problem generated by Li's

IIM and hence gives a second-order accurate solution to the partial differential equation.

The multigrid solution for Problem 1 with jumps in the coefficients β, the solution, the flux, and the source term, was obtained at a rate of .13 using V(2,2) with 2 levels, .06 using V(4,4) with 2-levels, and .06 using V(4,4) with 3-levels. For Problem 2, with a large jump in $[u_n]$, but $[\beta u_n] = 0$, we obtained rates of .03 for errors and .06 for residuals using V(4,4) and 2-levels.

In order to achieve such rates, a modified bilinear interpolation scheme that takes advantage of known jumps in the problem at the interface as well as knowledge of where the interface is located was developed. If the second derivatives in u (for continuation) or discrete error (for V-cycle) are not too big, this interpolation can be expected to give good results to $O(h^2)$. If a coarse grid cell is regular, then the modified interpolation reduces to ordinary bilinear interpolation, and restriction becomes full-weighting. For V-cycle interpolation, the assumption that $[e]_\Gamma = 0$ and $[\beta e_n]_\Gamma = 0$ seems to be a reasonable one since we achieved a factor of 7 to 10 improvement over the pre-smoothed result after doing coarse grid correction.

This multigrid approach was used successfully to generate pressures from which velocities were obtained for the groundwater flow application in Problem 3. The contaminant was advected in this velocity field using a **Clawpack** routine that did not know about the location of the interface. Results showed that the contaminant front was very sharp.

There are still many improvements that can be made or questions that should be answered. First, the coarse grid problems come directly from an immersed interface formulation on the given grid level, not from a Galerkin condition of the fine grid problem. It is possible that one could use even coarser grids if a Galerkin approach is used. In addition, a Galerkin formulation may be more amenable to different smoothing strategies than our approach and could be beneficial when more complicated problems are tackled. Second, we plan to compare this approach to the operator-induced interpolation approaches that others have taken. In particular, Dendy's Black Box solver for nonsymmetric problems, [9], could take the 6-point stencil generated by the IIM and infer an interpolation strategy, as well as automatically determining the coarser grids without explicit knowledge of the interface.

ACKNOWLEDGMENTS

The author would like to thank Randy LeVeque for useful discussions about the immersed interface method and for running the **Clawpack** routine for Problem 3. Steve McCormick and Tom Manteuffel have provided a stimulating environment at the Applied Math Program, University of Colorado, Boulder for learning about multigrid. Joel Dendy and Victor Brandy gave tremendous insight into multigrid approaches for problems with discontinuities. Thanks also to Paul Swartztrauber in the Scientific Computing Division at the National Center for Atmospheric Research for arranging financial support during my sabbatical year.

REFERENCES

[1] C. Tu and C.S. Peskin, *Stability and instability in the computation of flows with moving immersed boundaries: a comparison of three methods*, SIAM J. Sci. Stat. Comput., 13(1992), pp. 1361-1376.

[2] R.J. LeVeque and Z. Li, *Immersed interface methods for Stokes flow with elastic boundaries or surface tension*, available from `ftp://amath.washington.edu/pub/rjl/papers/rjl-li:stokes`

[3] C.S. Peskin, *Numerical analysis of blood flow in the heart*, J. Comput. Phys., 25(1977), pp. 220-252.

[4] Z. Li, *The immersed interface method - a numerical approach for partial differential equations with interfaces*, Ph.D. thesis, University of Washington, Department of Applied Mathematics, Seattle, WA.

[5] R.J. LeVeque and Z. Li, *The Immersed Interface Method for Elliptic Equations with Discontinuous Coefficients and Singular Sources*, SIAM J. Numer. Analy., 31(1994), pp. 1019-1044.

[6] L. Adams and Z. Yang, *A comparison of techniques for solving ill-conditioned problems arising from the immersed boundary method*, Proceedings of Symposia in Applied Mathematics, Vancouver, CA, Aug. 1993.

[7] R.E. Alcouffe, A. Brandt, J.E. Dendy, Jr., and J.W. Painter, *The multigrid method for the diffusion equation with strongly discontinuous coefficients*, SIAM J. Sci. Stat. Comp., 2(1981), pp. 430-454.

[8] J.E. Dendy, Jr., *Black box multigrid*, J. Comp. Phys., 48(1982), pp. 366-386.

[9] J.E. Dendy, Jr., *Black box multigrid for nonsymmetric problems*, App. Math. Comp., 13(1983), pp. 261-283.

[10] J.E. Dendy, Jr., *Multigrid methods for diffusion equations with highly discontinuous coefficients*, Trans. Amer. Num. Soc., 56(1988), p.20.

[11] J.E. Dendy, S.F. McCormick, J.W. Ruge, T.F. Russell, S. Schaffer, *Multigrid methods for three-dimensional petroleum reservoir simulation*, Proceedings of the Tenth Symposium on Reservoir Simulation, Houston, TX, Feb. 6-8, 1989, pp. 19-25.

[12] P.M. deZeeuw, *Matrix-dependent prolongations and restricitons in a blackbox multigrid solver*, J. Comput. Appl. Math., 3(1990), pp. 1-27.

[13] M. Khalil, P. Wesseling, *Vertex-centered and cell- centered multigrid for interface problems*, J. Comp. Phys., 1991.

[14] C. Liu, Z. Liu, and S. McCormick, *An efficient multigrid scheme for elliptic equations with discontinuous coefficients*, Technical report, Computational Mathematics Group, University of Colorado at Denver, 1991.

[15] A. Fogelson and J. Keener, Dept. of Math., University of Utah, private communication.

[16] R.J. LeVeque, *CLAWPACK: A software package for solving multi-dimensional conservation laws*, Proc. 5th Intl. Conf. Hyperbolic Problems, 1994., available from `ftp://amath.washington.edu/pub/rjl/papers/rjl:hyp94.ps.Z`.

[17] R.J. LeVeque, *CLAWPACK software*, available from `netlib.att.com` in `netlib/pdes/claw` or on the Web at the URL `ftp://amath.washington.edu/pub/rjl/programs/clawpack.html`.

Smoothers for Optimization Problems

Eyal Arian* and Shlomo Ta'asan†

Abstract

We present a multigrid one-shot algorithm, and a smoothing analysis, for the numerical solution of optimal control problems which are governed by an elliptic PDE. The analysis provides a simple tool to determine a smoothing minimization process which is essential for multigrid application. Numerical results include optimal control of boundary data using different discretization schemes and an optimal shape design problem in 2D with Dirichlet boundary conditions.

1 Introduction

In this work we use multigrid methods to accelerate the numerical solution of optimization problems governed by an elliptic PDE. The necessary conditions for a minimum are given as a set of three equations: state, costate and design. The state equation is a PDE which depends on the design variables. The costate equation is a PDE for the Lagrange multipliers and is of the same type as the state PDE. In an optimal shape design (OSD) problem the design variable is the position of the boundary therefore the design equation is defined only on the boundary.

Based on the necessary conditions for the minimum, the gradient of the cost-function with respect to the discrete design variables is given by the residuals of the design equation (assuming that the residuals of the state and costate equations are zero). A gradient based algorithm can then be constructed by an iterative method which solves sequentially the state and the costate equations and then updates the design variables with the gradients.

Multigrid (MG) methods can accelerate this process in various ways. In [1] A. Jameson used a MG cycle to solve the state and costate equations in an aerodynamic shape design problem. Later, a "one-shot" method was proposed by S. Ta'asan, [2], and applied to aerodynamic shape design by S. Ta'asan, G. Kuruvila and M. D. Salas [3, 4], which uses a few coarse grids for the optimization process, where the design variables are restricted to a finite dimensional design space which correspond to smooth solutions. In [5, 6, 7] the MG one-shot method was extended to the infinite design space in which the design variables are updated on all levels as originally suggested by A. Brandt [8]. The main difficulty there is to provide a minimization algorithm which smoothes the design variables. We present a simple Fourier analysis which estimates the smoothing of the minimization process and provides a tool to establish smoothers by preconditioning if needed.

Numerical examples include a linear optimal boundary control problem using different discretization schemes and a non-linear optimal shape design problem using a body-fitted grid. Results are given in two dimensions.

*ICASE, Mail Stop 132C, NASA Langley Research Center, Hampton, VA 23681
†Dept. of Mathematics, Carnegie-Mellon University, Pittsburgh, PA 15213

2 The Problem

We address two classes of optimization problems which are strongly related. One class is "optimal shape design" (OSD) problems where the shape of the domain, in which a PDE is solved, is the variable to be optimized with respect to the PDE solution. These are generally non-linear problems which arise in many applications. A simpler class of optimization problems is optimal control of boundary data in a fixed domain boundary value problem. These problems are related to OSD problems using the small disturbance approximation. In the following, formal definitions of the above are given.

2.1 Optimal Shape Design

Let $\mathcal{O}$ be a bounded set in $\mathbb{R}^d$, and let Ω be a close subdomain in $\mathcal{O}$. The problem is to find an optimal domain $\Omega^* \in \mathcal{O}$ and a "state variable", $\phi \in L_2(\Omega^*)$, subject to the " state equation", such that a given cost function, $F(\Omega, \phi(\Omega))$, defined on $\mathcal{O} \times L^2(\mathcal{O})$ will be minimized;

$$\min_{\Omega \in \mathcal{O}} F(\Omega, \phi(\Omega)) \tag{2.1}$$

where ϕ satisfies the following PDE

$$\begin{cases} L\phi = f & \text{on } \Omega \\ B\phi = 0 & \text{on } \partial\Omega \end{cases} \tag{2.2}$$

where L ($x \in \mathcal{O}$) is an elliptic differential operator of order $2m$ defined on $\mathcal{O}$ and B a boundary operator.

An extensive cover of OSD theory and references can be found also in [9, 10, 11].

2.2 The Small Disturbance Approximation

Consider a solution ϕ_0 of the state equation (2.2) in a domain Ω_0. Let Γ be part of the boundary, (depending on the problem), $\Gamma \subset \partial\Omega$, and consider a perturbation of the boundary position Γ with (see Fig.1):

$$\Gamma(x') \leftarrow \Gamma(x') + \varepsilon\tilde{\alpha}(x')\hat{n}(x'). \tag{2.3}$$

where $\hat{n}(x')$ is the normal to the boundary.

The perturbed boundary,$\Gamma_{\varepsilon\tilde{\alpha}}$, defines a domain, $\Omega_{\varepsilon\tilde{\alpha}}$, with a solution $\phi_{\varepsilon\tilde{\alpha}}$. The solution is extended analytically to a neighborhood containing $\Omega \bigcup \Omega_{\varepsilon\tilde{\alpha}}$. The extension is denoted as the original function. The following are relations between quantities on the perturbed domain, $\Omega_{\varepsilon\tilde{\alpha}}$, and those on the original one:

$$L\phi_{\varepsilon\tilde{\alpha}}\Big|_{\Omega_{\varepsilon\tilde{\alpha}}} = L\phi\Big|_{\Omega_0} + \varepsilon L_\phi(\phi)\tilde{\phi}\Big|_{\Omega_0} + O(\varepsilon^2) \tag{2.4}$$

$$B\phi_{\varepsilon\tilde{\alpha}}\Big|_{\Gamma_{\varepsilon\tilde{\alpha}}} = B\phi\Big|_{\Gamma_0} + \varepsilon B_\phi(\phi)\tilde{\phi}\Big|_{\Gamma_0} + \varepsilon B_\alpha(\phi)\tilde{\alpha}\Big|_{\Gamma_0} + O(\varepsilon^2) \tag{2.5}$$

The second order terms in (2.4) and (2.5) can be neglected for sufficiently small ε, depending on $\tilde{\alpha}$. For example assume $\tilde{\alpha}$ is composed of a Fourier frequency ξ, $\tilde{\alpha} = e^{i\xi x}$, then ε should be smaller than the wavelength $\frac{1}{\xi}$, i.e. $\varepsilon \ll \frac{1}{\xi}$.

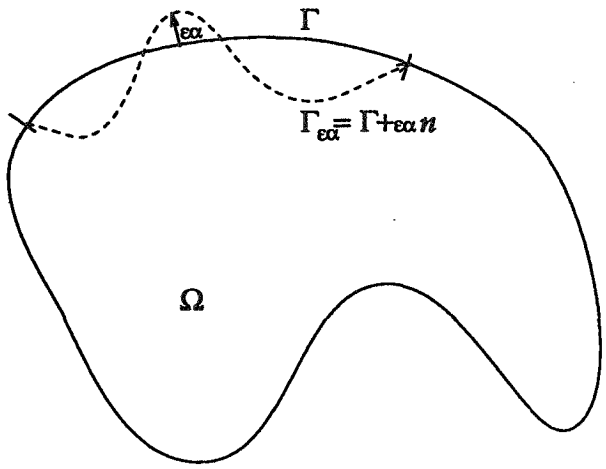


Figure 1: Perturbation of the boundary by $\varepsilon\tilde{\alpha}$ in the normal direction.

2.2.1 The Small Disturbance Minimization Problem

Relations (2.4) and (2.5) are used to reduce the OSD problem (2.1)-(2.2) to a minimization problem on a fixed domain with some unknown boundary data. In this problem Ω and ϕ are fixed, while $u = \tilde{\alpha}$ and $\varphi = \tilde{\phi}$ are the design and the state variables, respectively. The optimization problem is given by

$$\min_u \bar{F}(\varphi, u) \tag{2.6}$$

where φ satisfies

$$\begin{cases} L_\phi(\phi, u)\varphi = 0 & \text{on } \Omega \\ B_\phi(\phi)\varphi = -B_\alpha(\phi)u & \text{on } \Gamma. \end{cases} \tag{2.7}$$

2.3 Optimal Control of Boundary Data

The following problem is a more general formulation of the minimization problem which arises when performing the small disturbance approximation to optimal shape design problems.

Let Ω be a bounded open set of $\mathbb{R}^d$ with smooth boundary Γ and let ϕ be a real valued function on Ω. Let $\mathcal{U}$ and $\mathcal{W}$ be Hilbert spaces of real valued functions which are defined on Γ and Ω, respectively.

The problem is to find the "design variable", $u \in \mathcal{U}$, and the "state variable", $\phi \in \mathcal{W}$, such that a given cost function, $F(u, \phi(u))$, defined on $\mathcal{U} \times \mathcal{W}$, will be minimized. Here ϕ satisfies an elliptic PDE which is defined on Ω and will be referred to as the "analysis problem" or the "state equation":

$$\begin{array}{ll} \min_{u\in\mathcal{U}} F(u, \phi(u)) & \text{on } \Gamma \\ L(\phi, u) = 0 & \text{on } \Omega \end{array} \tag{2.8}$$

3 Derivation of the Necessary Conditions for a Minimum

We apply the adjoint method to the optimal boundary control problem (2.8). The variable space is enlarged by adding Lagrange multiplier functions or costate variables denoted by λ. A Lagrangian is

defined to be the sum of the cost-function and a linear term in the costate variables which vanishes as the constraint equation is satisfied;

$$E(\phi, \lambda, u) = F(u, \phi) - \langle \lambda, L(\phi, u) \rangle . \tag{3.1}$$

A perturbation of the Lagrangian with respect to all the variables independently, i.e., state, costate and design, results in a variation of the Lagrangian:

$$\begin{array}{l} \phi \leftarrow \phi + \varepsilon \tilde{\phi} \\ \lambda \leftarrow \lambda + \varepsilon \tilde{\lambda} \\ u \leftarrow u + \varepsilon \tilde{u} \end{array} \tag{3.2}$$

with $\tilde{\phi}, \tilde{\lambda} \in L_2(\Omega)$, $\tilde{u} \in \mathcal{U}$ and ε is a small real parameter. The variation of the Lagrange function, δE, in the first order approximation in ε, is given in the following form:

$$\delta E = -\varepsilon \left\langle \tilde{\phi}, L_\phi^*(\phi, u)\lambda + F_\phi \right\rangle - \varepsilon \left\langle \tilde{\lambda}, L(\phi, u) \right\rangle + \varepsilon \left\langle \tilde{u}, L_u^*(\phi, u)\lambda + F_u \right\rangle \tag{3.3}$$

where L_ϕ^* and L_u^* are the adjoint operators of L_ϕ and L_u, respectively. The requirement that the first approximation terms vanish results in the necessary condition for a minimum which will be referred to as the state, the costate, and the design equations:

$$\begin{array}{ll} \underline{State:} & L(\phi, u) = 0 \\ \underline{Costate:} & L_\phi^*(\phi, u)\lambda + F_\phi(\phi, u) = 0 \\ \underline{Design:} & L_u^*(\phi, u)\lambda + F_u(\phi, u) = 0. \end{array} \tag{3.4}$$

From here on we will use the notation $\mathcal{A}(u)$ for the design equation residual, i.e.,

$$\mathcal{A}(u) = -L_u^*(\phi(u), u)\lambda(u) - F_u(\phi(u), u) \tag{3.5}$$

where $\phi(u)$ and $\lambda(u)$ in (3.5) are solutions of the state and costate equations.

The application of the adjoint method to optimal shape problems is performed in a similar manner [5, 7].

4 Discretization

When discretizing the problem it is possible either to derive the necessary conditions for a minimum in the continuous formulation and then discretize or to discretize the cost-function together with the state equation and then derive the discrete necessary conditions. In the latter case the discrete minimization problem is given by:

$$\begin{array}{ll} \min_{u^h} F^h(u^h, \phi^h) & on\ \Gamma^h \\ L^h(\phi^h, u^h) = 0 & on\ \Omega^h. \end{array} \tag{4.1}$$

As the grid mesh size, h, goes to zero, solutions of both approaches should converge to the differential solution. However, for a finite mesh size, discretization and necessary conditions do not necessarily commute. The solutions of both should be within the discretization error of the differential solution. In this paper we used both strategies. In the optimal boundary control problem, the derivation of the necessary conditions for a minimum was done in the continuous space, and then these conditions

were discretized, while in the optimal shape problem the necessary conditions for a minimum were derived in the discrete space. The discrete state, costate and design equations are:

$$\begin{array}{ll} L^h(\phi^h, u^h) = 0 & on\ \Omega^h \\ L^{h*}_{\phi}(\phi^h, u^h)\lambda^h + F^h_{\phi}(\phi^h, u^h) = 0 & on\ \Omega^h \\ L^{h*}_{u}(\phi^h, u^h)\lambda^h + F^h_{u}(\phi^h, u^h) = 0 & on\ \Gamma^h. \end{array} \tag{4.2}$$

We define $A^h(u^h)$ similarly to (3.5).

5 A Gradient Descent Algorithm

If the state and costate equations are satisfied then the gradient of the cost-function with respect to the design variables is given by the residuals of the design equation (see [5, 6, 7]):

$$\nabla_u F = \mathcal{A}(u).$$

The following is a gradient descent minimization algorithm which follows immediately from the above.

1. Start with an initial approximation for the design variables, u^h_0.
2. Solve the state equation for ϕ^h.
3. Solve the costate equation for λ^h.
4. Compute the amplitude of the perturbation, β, with a line search, and update the design variables: $u^h \leftarrow u^h + \beta\ \mathcal{A}^h(u^h)$.
5. If the residuals of the state, the costate and the design equations are greater than some preassigned value, in L_2 norm, then go to 2; else stop.

Note that steps 2, 3 and 5 consist of a global computation over the whole domain.

The complexity of this algorithm is given by $O(M^p N^l)$, where M is the number of design parameters, N is the number of grid points, and p and l are integers which depend on the problem and the PDE solver which is used to solve the state and costate equations. For example, if a MG solver is used to solve the PDE then $l = 1$.

6 Relaxation of the Design Variables in a Multigrid Cycle

The Full Approximation Scheme (FAS) is used to represent the state, the costate and the design equations on coarser grids. On each level a relaxation is performed on the state, costate and design variables. The state and costate equations, which are elliptic PDE, are relaxed by a Gauss-Seidel or damped Jacobi relaxations. The design variables are relaxed by

$$u^h \leftarrow u^h + \beta^h \mathcal{F}^h \mathcal{A}^h(u^h), \tag{6.1}$$

where β^h and $\mathcal{F}^h$ are chosen to guarantee good smoothing for the design variables and where $\mathcal{A}^h(u^h)$ are the residuals of the design equation. The choice of $\mathcal{F}^h$ is discussed in Sec.8. This step should be followed by an update of the state and costate solutions. The construction of β^h and $\mathcal{F}^h$ is done so that the boundary data is updated with a high frequency dominated quantity.

6.1 Ellipticity and Computational Cost

In elliptic systems a perturbation of the boundary condition with a Fourier mode $e^{i\omega x}$ has an exponentially decaying effect on the interior solution of the form $e^{-\sigma(\omega)y}$, where y is the distance from the boundary and $\sigma(\omega)$ is a positive monotonically increasing function of ω for large $|\omega|$, [12]. For the Laplace equation the decaying rate is given by $e^{-|\omega|y}$. Therefore, in an MG scheme it is preferable to perturb the boundary condition with only high frequency modes relative to the given level. In that case only local relaxations will be needed in order to update the solutions after each optimization step. Also in non-linear problems the line search procedure, which calculates the amplitude of the minimization step (β), requires a trial perturbation of the boundary. As a result of the local effect of such a perturbation the computational cost of the minimization step is only $O(\sqrt{N})$ operations (in two dimensions).

On the coarsest grid the relaxation of the design variables is given in Eqn.(6.1) with $\mathcal{F} = I$ thus taking into account the lowest frequencies. In that case the state and costate PDE are solved over the whole domain.

7 Smoothing Analysis

The Fourier analysis is based on calculating the symbol of the transformation between errors in the design variables and residuals of the design equation. The analysis can be done either in the continuous or discrete level. In the following we present the continuous analysis. The advantage of performing the analysis at the continuous level is the elimination of the effect of specific discretization on the above transformation. One objective of the analysis is to determine if the problem is well posed in the sense that small changes in the residuals of the design equation correspond to small changes in the design variables.

7.1 Reduction to the Boundary

The analysis is done by considering the high frequency errors in the design variables in half space (Fig.2). Then with a standard procedure the problem in half space is reduced to the boundary [13].

We assume that the state and costate equations are satisfied when the design variables are updated. Another assumption is that in the vicinity of the boundary, the non-smooth errors can be analyzed using a half space geometry. This approximation is valid since in elliptic problems non-smooth Fourier modes decay exponentially into the interior. For simplicity the analysis is performed for a second order elliptic equation in a two dimensional space.

Consider a two dimensional geometry where the x axis is parallel to the boundary and the y axis is in the normal direction (see Fig.2). We want to study the mapping from errors in the design variables to the residuals of the design equation. The errors of the state and costate variables satisfy a homogeneous equation in the interior. The state, the costate, and the design errors are given by

$$\begin{aligned} \bar{\phi}(x,y) &= \int_{-\infty}^{\infty} \hat{\phi}(\omega) e^{i\omega x} e^{-\sigma(\omega)y} d\omega \\ \bar{\lambda}(x,y) &= \int_{-\infty}^{\infty} \hat{\lambda}(\omega) e^{i\omega x} e^{-\sigma(\omega)y} d\omega \\ \bar{u}(x) &= \int_{-\infty}^{\infty} \hat{u}(\omega) e^{i\omega x} d\omega \end{aligned} \tag{7.1}$$

where $Le^{i\omega x}e^{-\sigma(\omega)y} = 0$ and $\sigma(\omega) > 0$ (for the Laplace operator $\sigma(\omega) = |\omega|$). By substituting these expressions into the boundary conditions of the state and costate error equations, we obtain relations

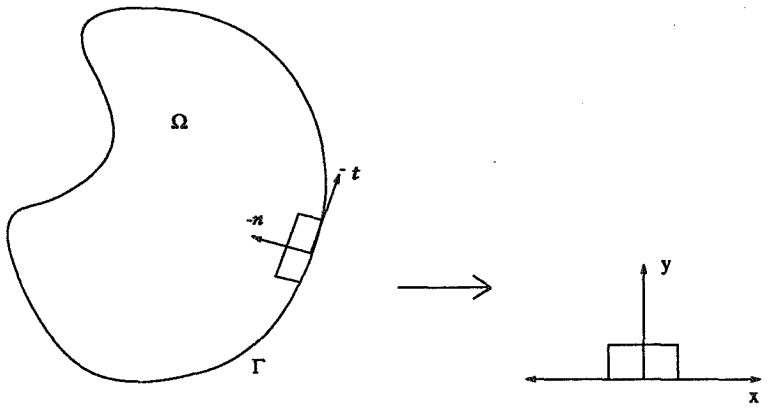


Figure 2: A vicinity of a point on the boundary is transformed into a half space geometry.

between $\hat{\phi}(\omega), \hat{\lambda}(\omega)$ and $\hat{u}(\omega)$. ¿From the set of boundary conditions for the state and the costate equations and from the design equation (which is defined on the boundary) we can deduce a relation between the errors in the design variables and the residuals of the design equation;

$$\hat{\mathcal{A}}(\omega) = \hat{T}(\omega)\hat{u}(\omega). \tag{7.2}$$

$\hat{T}(\omega)$ is the symbol of the Hessian of the cost function, F, subject to the PDE constraint. In this work we use this symbol to estimate the smoothing properties of the minimization procedure. If the symbol of the transformation $\hat{T}(\omega)$ is a monotonically decreasing function in ω then one expects that the relaxation of the design variable will not be a good smoother. On the other hand if $\hat{T}(\omega)$ is a monotonically increasing function in ω, for large $|\omega|$, then high frequency errors in the shape are amplified in the residuals of the design equation and good smoothing of the minimization process is anticipated. Note that this analysis deals only with the high-frequencies.

7.2 Analysis of Optimal Shape Design Problems

The optimal shape problem is reduced of an optimal control of boundary data problem by the small disturbance approximation as explained in Sec. 2.2. However, in this case the resulting equations have variable coefficients and a more delicate analysis is required. This is done by freezing the coefficients, at a point x_0, which is justified as long as the changes in the design variables are highly oscillatory compared to changes in the coefficients appearing in the small disturbance problem. As a result of such an analysis one obtains the transformation between errors in the shape variables and the residuals of the design equation in the neighborhood of x_0:

$$\hat{\mathcal{A}}(\omega, a_0) = \hat{T}(\omega, a_0)\hat{\alpha}(\omega). \tag{7.3}$$

where a_0 stands for a quantity which is computed at x_0.

8 Construction of a Smoothing Minimization Process

We are interested in the amplification factor of the error in the design variables as a result of the multigrid minimization process. The relations between the errors in the design variables before and after the relaxation are followed from Eqns.(6.1) and (7.2):

$$\hat{u}^h_{new} = \hat{R}^h(\theta)\hat{u}^h_{old} \tag{8.1}$$

where the relaxation symbol $\hat{R}^h(\theta)$ is given by:

$$\hat{R}^h(\theta) = 1 + \beta^h \hat{\mathcal{F}}^h \hat{T}^h(\theta). \tag{8.2}$$

For multigrid purposes it is desirable that $\hat{R}^h(\theta)$ has small values in the high frequency range ($\frac{\pi}{2} \leq |\theta| \leq \pi$). If this is the case, the relaxation will reduce effectively the high frequency errors of the design variables prior to restricting their values to the coarse grid.

Choice of Preconditioner

In some cases the relaxation without the use of a preconditioner, $\mathcal{F}^h$, does not smooth the errors effectively for any choice of β^h. In these cases preconditioning of the design residuals is required. If chosen properly the symbol $\hat{\mathcal{F}}^h(\theta)\hat{T}^h(\theta)$ is dominated by the high frequencies and a proper choice of β^h will result in good smoothing. The preconditioner is particularly effective for problems in which the transformation $\hat{T}^h(\theta)$ is a monotonically decreasing function which has small values in the high frequencies. In these cases the minimization process does not smooth the errors effectively, and therefore without the use of a proper preconditioner, high-frequency oscillatory errors in the design variables are slow to converge, and in some cases might result in the divergence of the algorithm. Computational experiments using preconditioning were reported in [5, 6].

Evaluation of the optimization step size β^h

In a multigrid cycle the relaxation should be effective mainly in the high frequency range. The relaxation parameter β^h is chosen such that the maximum of $|\hat{R}^h(\theta)|$ in the high frequencies will be minimal, that is,

$$\min_{\beta^h} \max_{\frac{\pi}{2} \leq |\theta| \leq \pi} |1 + \beta^h \hat{\mathcal{F}}^h \hat{T}^h(\theta)|. \tag{8.3}$$

One can show that if the symbol $\hat{T}^h(\theta)$ does not change sign β^h is given by

$$\beta^h = -\frac{2}{(\hat{\mathcal{F}}^h\hat{T}^h)_{min} + (\hat{\mathcal{F}}^h\hat{T}^h)_{max}} \tag{8.4}$$

where $(\hat{\mathcal{F}}^h\hat{T}^h)_{min}$ and $(\hat{\mathcal{F}}^h\hat{T}^h)_{max}$ are the minimal and maximal values of $\hat{\mathcal{F}}^h(\theta)\hat{T}^h(\theta)$ in the range $(\frac{\pi}{2} \leq |\theta| \leq \pi)$. In most of the practical problems the symbol $\hat{\mathcal{F}}^h\hat{T}^h(\theta)$ is monotonous, thus β^h is given by

$$\beta^h = -\frac{2}{\hat{\mathcal{F}}^h(\frac{\pi}{2})\hat{T}^h(\frac{\pi}{2}) + \hat{\mathcal{F}}^h(\pi)\hat{T}^h(\pi)}. \tag{8.5}$$

In this way the size of the minimization step amplitude, β^h, is found by Fourier analysis instead of using a line-search, thus reducing the computational cost of each optimization step. However, this was demonstrated in practice only for linear problems (see Sec. 9.1.2).

9 Numerical Examples

We give two numerical examples: an optimal control of Dirichlet data with a fixed geometry and an optimal shape design problem where the position of the boundary is the design variable. The purpose of the first example is to demonstrate the use of the smoothing analysis to choose the best discretization for a given problem given a choice between a few possibilities. It is shown both analytically and numerically that different discretizations result in different smoothing rates of the minimization process. The purpose of the optimal shape design example is to demonstrate the effectiveness of our method.

9.1 Dirichlet Boundary Control Problem

The minimization problem is defined by

$$\min_{u(x)} \int_{y=1} \left(\frac{\partial \phi}{\partial n} - f^*(x)\right)^2 dx + \eta \int_{y=1} u^2 dx \tag{9.1}$$

where η is a fixed non-negative parameter, $f^*(x)$ is a given function and where ϕ satisfies the state equation. The state equation is given by

$$\begin{cases} \triangle \phi = f & \text{on } \Omega \\ \phi = u(x) & \text{on } y = 1 \\ \phi = \phi_0 & \text{on } y = 0 \end{cases} \tag{9.2}$$

where $\Omega = \{0 < x < 1 \ ; \ 0 < y < 1\}$ and periodicity is assumed in the x direction. The costate equation is given by

$$\begin{cases} \triangle \lambda = 0 & \text{on } \Omega \\ \lambda + 2(\frac{\partial \phi}{\partial n} - f^*(x)) = 0 & \text{on } y = 1 \\ \lambda = 0 & \text{on } y = 0 \end{cases} \tag{9.3}$$

The design equation is given by

$$\mathcal{A} = \frac{\partial \lambda}{\partial n} - 2\eta\phi = 0 \quad \text{on } y = 1. \tag{9.4}$$

9.1.1 Discretization

The state, the costate, and the design equations are discretized in four different ways. In three discretizations all the unknowns are defined on the vertices of the grid lines as shown in Fig.3A (will be referred to as the "vertex grid"). The control variables are defined on the intersections of the grid points with the boundary. The normal derivative in the cost function was approximated with a first (VX1), a second (VX2) order approximation, and with the use of virtual points outside the domain (VX3). The fourth discretization is cell-centered (CC), where the variables are defined on the centers of the grid cells as shown in Fig.3B. The grid is extended out of the domain and virtual cell centered points are defined neighboring (exterior of) the domain. A Dirichlet boundary condition is given for the average of the variables neighboring the boundary. The design variables are defined on the centers of the segments connecting the intersection of the grid with the boundary. Note that in the multigrid scheme, the vertices of the grids on different scales are nested while in the cell-center case the cells are nested.

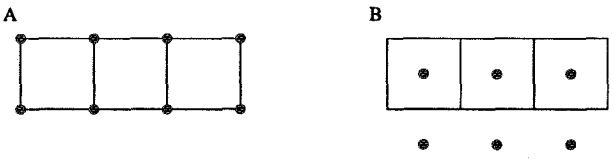


Figure 3: Vertex (A) and cell centered (B) grids.

The different approximations for the normal derivative on the boundary are:
1) A first order approximation for the normal derivative

$$VX1: \quad \frac{\partial \phi}{\partial n_i} = \frac{\phi_{i,2} - \phi_{i,1}}{h}, \tag{9.5}$$

2) A second order approximation for the normal derivative

$$VX2: \quad \frac{\partial \phi}{\partial n_i} = \frac{-\frac{3}{2}\phi_{i,1} + 2\phi_{i,2} - \frac{1}{2}\phi_{i,3}}{h}, \tag{9.6}$$

3) A use of a virtual point out of the domain (where its value is determined with the application of the interior operator on the boundary)

$$VX3: \quad \frac{\partial \phi}{\partial n_i} = \frac{\phi_{i,1} - \phi_{i,-1}}{2h}. \tag{9.7}$$

4) A cell centered discretization

$$CC: \quad \frac{\partial \phi}{\partial n_i} = \frac{\phi_{i,\frac{1}{2}} - \phi_{i,-\frac{1}{2}}}{h}. \tag{9.8}$$

9.1.2 Analysis: Reduction to the Boundary

In the following the design equation for the Dirichlet boundary control problem is analyzed in the discrete space. The second order finite difference approximation of the Laplacian (which was used in the numerical experiments) is given by

$$\triangle^h = \frac{1}{h^2} \begin{pmatrix} & 1 & \\ 1 & -4 & 1 \\ & 1 & \end{pmatrix}. \tag{9.9}$$

The term $e^{\sigma(\theta)}$, which is the discrete analog of $\sigma(\omega)$ in Eqn.(7.1), satisfies the following second order equation

$$e^{\sigma(\theta)} + (-4 + 2\cos\theta) + e^{-\sigma(\theta)} = 0. \tag{9.10}$$

In order to calculate the Fourier symbol of the design equation (9.4), the symbol of the normal derivatives (9.5)-(9.8) is given first.

The Fourier Symbol of the Normal Derivatives

$$VX1: \quad \frac{\hat{\partial}^h}{\partial n}(\theta) = \frac{e^{-\sigma(\theta)} - 1}{h} \tag{9.11}$$

$$VX2: \quad \frac{\hat{\partial}^h}{\partial n}(\theta) = \frac{-\frac{1}{2}e^{-2\sigma(\theta)} + 2e^{-\sigma(\theta)} - \frac{3}{2}}{h} \tag{9.12}$$

$$VX3: \quad \frac{\hat{\partial}^h}{\partial n}(\theta) = \frac{e^{-\sigma(\theta)} - e^{\sigma(\theta)}}{2h} \tag{9.13}$$

$$CC: \quad \frac{\hat{\partial}^h}{\partial n}(\theta) = \frac{e^{-\frac{1}{2}\sigma(\theta)} - e^{\frac{1}{2}\sigma(\theta)}}{h}. \tag{9.14}$$

The Fourier Symbol of the Design Equation

In terms of the normal derivatives, the transformation $\hat{T}^h(\theta)$ (see Eqn.(9.4)) is given by

$$\hat{T}^h(\theta) = -2\Big[\Big(\frac{\hat{\partial}^h}{\partial n}(\theta)\Big)^2 + \eta\Big]. \tag{9.15}$$

As the parameter η increases the weight of the low frequencies is increased relative to the high frequencies.

The amplitude of the minimization step, β^h, given in Eqn.(8.5) is reduced to

$$\beta^h = \frac{1}{\Big(\frac{\hat{\partial}^h}{\partial n}(\frac{\pi}{2})\Big)^2 + \Big(\frac{\hat{\partial}^h}{\partial n}(\pi)\Big)^2 + 2\eta}. \tag{9.16}$$

In Fig.5 the relaxation symbol $\hat{R}^h(\theta) = 1 + \beta^h\hat{T}^h(\theta)$ is plotted for the above four discretizations. For all four discretizations the relaxation reduces the high frequency errors by a factor smaller than 0.5.

Fig.6 depicts the maximal eigenvalue, $|\lambda|_{max}$, of the two level convergence matrix as a function of the number of minimization steps, ν, on a given level. The factor by which the error is reduced as a result of a two level multigrid cycle is bounded by $|\lambda|_{max}$. It is implied by Fig.6 that the cell-centered (CC) and second order vertex (VX2) schemes are expected to have a better performance than the other vertex schemes.

9.1.3 Convergence Performance

In the numerical tests the problems (9.1)-(9.2) were solved for the four discretizations (9.5)-(9.8). In all of these problems there was no need to use a preconditioner, $\mathcal{F}$, since the transformation $\hat{T}^h(\theta)$ is dominated by the high frequencies. The minimization step amplitude, β^h, given by Eqn.(9.16) was used in the computations. The multigrid one shot algorithm was tested using between two and seven levels (Fig. 4). In all the tests the residuals of the state, the costate and the design equations were computed in L_2 norm.

In the two levels test (table 1), the finest grid was composed of $2^7 \times 2^7$ grid points and the coarsest grid was composed of $2^6 \times 2^6$ grid points. The parameter η in the cost function (9.1) was set to zero. In table 1 the two level analysis and the actual convergence rates are compared for the four different discretizations. The agreement between the predicted and actual convergence is well apparent.

In the multilevel test the fine grid was composed of $2^m \times 2^m$ points, with $m = 5, 6, 7$, and the coarsest grid was composed of 2×2 grid points. The tests with different choices of m were done in order to check if the algorithm is mesh-size dependent. All the results in Fig.4 were performed with a cell-centered discretization. Since the case $\eta = 0$ in (9.1) corresponds to a trivial minimization problem, the case $\eta = 1$ was tested, although in principle these cases are not different.

In all problems the error was reduced in each V-cycle by an order of magnitude, where each V-cycle has a computation complexity of $O(N)$ operations.

9.2 An Optimal Shape Design Problem in 2D

Problem Definition

The minimization problem is defined by

$$\min_{\Gamma(x)} \int_{\Gamma(x)} \Big(\frac{\partial \phi}{\partial y} - f^*(x)\Big)^2 dx \tag{9.17}$$

with $f^*(x)$ a given target function and where ϕ satisfies the state equation. The state and costate equations are given by

$$\begin{cases} \triangle \phi = f & \text{on } 0 < x < 1 \ ; \ \Gamma(x) < y < 1 \\ \phi = g(x) & \text{on } y = 1 \\ \phi = \phi_0 & \text{on } y = \Gamma(x). \end{cases} \tag{9.18}$$

The coordinates of grid points on the boundary $\Gamma = \Gamma^h(x)$ define the design variables. The initial approximation for the shape, on the coarsest grid, was a flat surface: $\Gamma^h(x) = 0$.

The costate equation

$$\begin{cases} \triangle \lambda = 0 & \text{on } 0 < x < 1 \ ; \ \Gamma(x) < y < 1 \\ \lambda = 0 & \text{on } y = 1 \\ \lambda + 2\cos^2\theta(\frac{\partial \phi}{\partial y} - f^*) = 0 & \text{on } y = \Gamma(x) \end{cases} \tag{9.19}$$

The design equation

The design equation is simplified by using the costate boundary condition yielding

$$\mathcal{A}(\phi, \lambda, \theta) = 0 \quad on \ \ y = \Gamma(x) \tag{9.20}$$

where

$$\mathcal{A}(\phi, \lambda, \theta) = -\frac{\partial \phi}{\partial y}\frac{\partial}{\partial \sigma}\big(\lambda \tan\theta\big) - \frac{\lambda}{\cos\theta}\frac{\partial^2 \phi}{\partial y^2} + \frac{\partial \phi}{\partial y}\frac{\partial \lambda}{\partial n}. \tag{9.21}$$

Smoothing Analysis

The smoothing analysis was done by first performing the small disturbance approximation resulting in a fixed domain minimization problem (see Sec. 2.2) and then reducing the problem to the boundary as is explained in Sec. 7. The result is given by the following mapping:

$$\hat{T}(\omega) = 2\Big(\frac{\partial \phi}{\partial y}(x_0)\Big)^2 \cos\theta_0 \big[i \sin(2\theta_0)\omega|\omega| + \cos(2\theta_0)|\omega|^2\big] + O(\omega).$$

For large ω there exists a positive constant C such that

$$|\hat{T}(\omega)| \geq C|\omega|^2 \tag{9.22}$$

thus high frequency errors in the shape are amplified in the residuals of the design equation. It is this type of problems which is difficult to solve numerically by a single grid algorithm and for which multigrid is an ideal accelerator.

Convergence Performance

The numerical test was done with one application of an FMG algorithm with 2 preliminary cycles, 4 optimization cycles per level and 10 relaxations on the coarsest level (the cycle which was used is $W(2,1)$). The line search uses 10 local relaxations which are performed on the four adjacent to the boundary grid lines (on all levels). The depicted residuals are the final residuals.

Tables 2 and 3 give the convergence rate of the OSD Dirichlet problem for different mesh-sizes and different curvatures of the target geometry: table 2 corresponds to the case $f^*(x) = 0.05\sin(2\pi x)$ and table 3 to the case $f^*(x) = 0.2e^{-30(x-0.5)^2}$. In both tables r_x, r_p and r_u correspond to the residuals of the state, the costate and the design equations, respectively, and $u - u_{exact}$ is the error in the design variables. In both cases the initial geometry was a flat boundary ($\Gamma(x) = 0$). The results show a mesh-size independent convergence rate for both cases.

Numerical experiments show that one application of a FMG scheme, with *four* cycles per level, was enough to reach the discretization error on all levels.

A Note on the Cell Centered Finite Volume Discretization

In problem (9.17)-(9.18), using a cell centered discretization, the transformation $\hat{T}^h(\theta)$ vanishes for the highest frequency, $\theta = \pi$, resulting in high frequency errors in the shape variables. Therefore we argue that for this problem a vertex grid is a preferable discretization.

Consider a flat boundary (x axis) and a boundary perturbation in the y direction of the form $\tilde{\Gamma}_i = \epsilon(-1)^i$ where the index i stands for the ith point on the boundary, and ϵ is some number smaller than the mesh size. As a result of such a perturbation the cell center position will not change since the cell center coordinate is the average of the vertices coordinates. Therefore the solutions of the state, costate and design equations will not detect the perturbation. In order to avoid the oscillatory errors from entering the boundary a penalty on the cost function, or a preconditioner on the design equation residuals, should be applied.

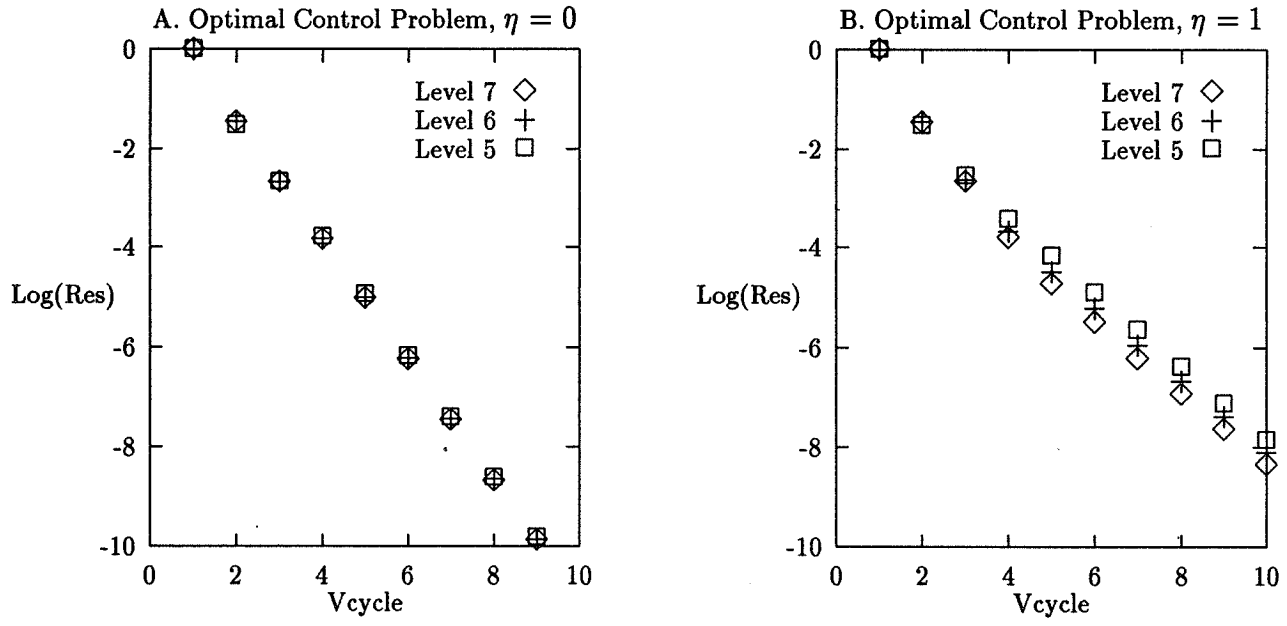


Figure 4: Convergence rates. A and B depict the Dirichlet boundary control problem with $\eta = 0$ and $\eta = 1$ respectively. The depicted residuals in A and B are the average of the computed state, costate, and design equations residuals.

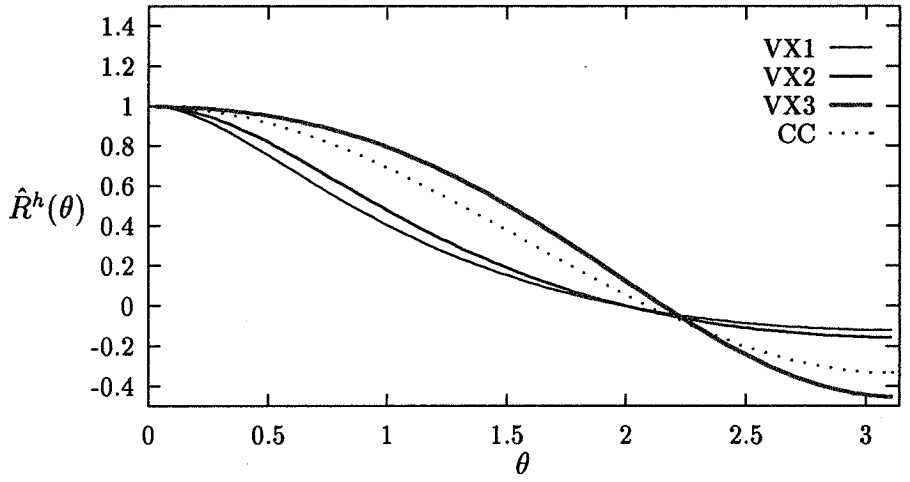


Figure 5: The symbol of the design variable relaxation for the Dirichlet boundary control problem with $\eta = 0$. Each curve represents a different discretization.

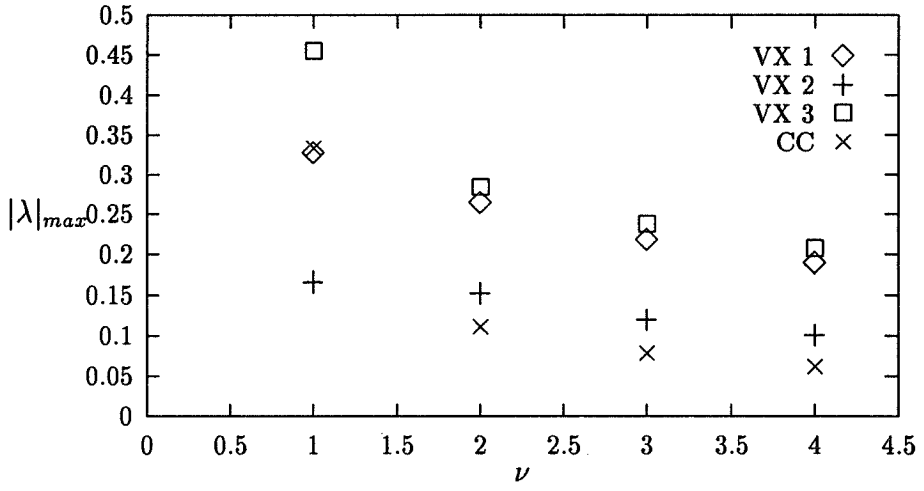


Figure 6: Two level analysis of asymptotic convergence rates, $|\lambda|_{max}$, as a function of the number of optimization steps, ν, for $\eta = 0$. Each symbol represents a different discretization.

	VX1		VX2		VX3		CC	
ν	TLA	NUM	TLA	NUM	TLA	NUM	TLA	NUM
1	0.327	0.320	0.166	0.166	0.454	0.439	0.333	0.276
2	0.264	0.255	0.153	0.152	0.283	0.278	0.111	0.105
3	0.218	0.188	0.120	0.100	0.236	0.229	0.078	0.067
4	0.189	0.181	0.101	0.080	0.206	0.202	0.061	0.035

Table 1: Two Level Analysis (TLA) versus tested (NUM) convergence rates for the optimal control of Dirichlet data problem, for various number of optimization steps, ν, on the fine level.

level	$\|r_x\|_2$	$\|r_p\|_2$	$\|r_u\|_2$	$\|u - u_{exact}\|_2$
2	$0.103e{-}15$	$0.872e{-}16$	$0.676e{-}07$	$0.142e{-}00$
3	$0.135e{-}03$	$0.713e{-}03$	$0.504e{-}03$	$0.674e{-}01$
4	$0.431e{-}04$	$0.283e{-}03$	$0.249e{-}03$	$0.326e{-}01$
5	$0.128e{-}04$	$0.429e{-}04$	$0.855e{-}04$	$0.157e{-}01$
6	$0.258e{-}05$	$0.967e{-}05$	$0.322e{-}04$	$0.744e{-}02$
7	$0.445e{-}06$	$0.217e{-}05$	$0.123e{-}04$	$0.374e{-}02$

Table 2: Convergence rates for the optimal shape design problem with a target distribution given by $f^*(x) = 0.05\sin(2\pi x)$

level	$\|r_x\|_2$	$\|r_p\|_2$	$\|r_u\|_2$	$\|u - u_{exact}\|_2$
2	$0.349e{-}15$	$0.757e{-}16$	$0.218e{-}07$	$0.318e{-}01$
3	$0.673e{-}03$	$0.901e{-}03$	$0.713e{-}03$	$0.505e{-}01$
4	$0.281e{-}03$	$0.112e{-}02$	$0.334e{-}02$	$0.371e{-}01$
5	$0.337e{-}04$	$0.286e{-}03$	$0.100e{-}02$	$0.216e{-}01$
6	$0.156e{-}04$	$0.156e{-}03$	$0.757e{-}03$	$0.655e{-}02$
7	$0.165e{-}05$	$0.335e{-}04$	$0.485e{-}03$	$0.381e{-}02$

Table 3: Convergence rates for the optimal shape design problem with a target distribution given by $f^*(x) = 0.2e^{-30(x-0.5)^2}$

References

[1] A. Jameson. Aerodynamic Design Via Control Theory, ICASE report NO. 88-64, November (1988), Journal of Scientific Computing. 3:233-260 (1988).

[2] S. Ta'asan. "One Shot" Methods for Optimal Control of Distributed Parameter Systems I: Finite Dimensional Control, ICASE report NO. 91-2, January (1991).

[3] S. Ta'asan, G. Kuruvila, M. D. Salas. Aerodynamic Design and Optimization in One Shot, 30th Aerospace Sciences Meeting & Exhibit, AIAA 92-0025, Jan. (1992).

[4] G. Kuruvila, S. Ta'asan, M. D. Salas. Airfoil Optimization by the One-Shot Method, Optimum Design Methods in Aerodynamics, AGARD-FDP-VKI Special Course, April 25-29 (1994).

[5] E. Arian. Multigrid Methods for Optimal Shape Design Governed by Elliptic Systems, Ph.D. Thesis, The Weizmann Institute of Science, Israel (1994).

[6] E. Arian and S. Ta'asan. Multigrid One Shot Methods for Optimal Design Problems: Infinite Dimensional Control, ICASE report No. 94-52 (1994).

[7] E. Arian, and S. Ta'asan. Shape Optimization in One Shot, Optimal Design and Control, Edited by J. Boggaard, J. Burkardt, M. Gunzburger, J. Peterson, Birkhäuser Boston Inc. (1995).

[8] A. Brandt. Multigrid Techniques: 1984 Guide, with Applications to Fluid Dynamics.

[9] O. Pironneau. Optimal Shape Design for Elliptic Systems, Springer Series in Computational Physics (1983).

[10] J. Haslinger and P. Neittaanmäki. Finite Element Approximation for OSD: Theory and Application, hon Wiley & Sons (1988).

[11] J. Sokolowski and J. Zolesio. Introduction to Shape Optimization, Springer-Verlag (1992).

[12] S. Agmon, A. Douglis, L. Nirenberg. Estimates Near the Boundary for Solutions of Elliptic Partial Differential Equations Satisfying General Boundary Conditions. Communications on Pure and Applied Math., Vol. XII 623-727 (1959).

[13] A. Calderon. Boundary Value Problems for Elliptic Equations, Proceedings of the Joint Soviet-American Symposium on Partial Differential Equations. Novosibirsk Acad. Sci. USSR 1-4, (1963).

MULTIGRID WITH OVERLAPPING PATCHES

Markus Berndt[1]
University of Colorado
Campus Box 526
Boulder, CO 80309-0526

Kristian Witsch[2]
Universität Düsseldorf
Universitätsstrasse 1
D-40225 Düsseldorf, Germany

SUMMARY

Solving boundary value problems with optimal efficiency requires adaptivity and multilevel techniques. In [6] an implementation of the AFACx algorithm (cf. [8]) is presented that is based on rectangular Cartesian grids. This implementation does not allow for the overlap of grids that lie on the same level of refinement. We investigate the case in which these grids overlap. A standard technique for overlapping grids is the Schwarz algorithm (cf. [12] and [13]). Some ways of using the Schwarz algorithm in a standard multigrid scheme are presented. Also, a problem that arises in some situations with non-aligned, overlapping grids is described. This situation comes up in a natural way when the Schwarz algorithm is used as a relaxation scheme within a multilevel algorithm. We identify the reason for the bad convergence and show that by more sophisticated interpolation the difficulties can be overcome. Then we present a multiplicative Schwarz algorithm for a large number of grids that has a high potential for parallelization. Finally we give some numerical results for the FACx algorithm with overlapping grids on each refinement level. The implementation of the described codes uses C++ and the array class libraries A++ and P++ (cf. [4], [5], and [11]). Using the A++/P++ programming environment, it was possible to move from a serial code to a parallel code within a few days.

[1]e-mail: Markus.Berndt@Colorado.EDU
[2]e-mail: witsch@numerik.uni-duesseldorf.de

INTRODUCTION

The Schwarz algorithm is a useful tool when it comes to adaptive refinement. The implementation of these complex algorithms can be kept simple by using regular grid structures for the discretization. However, any sophisticated refinement strategy will yield highly irregular refinement regions. In [5] and [10] an implementation of the AFACx algorithm is presented. This implementation is based on block structured refinement grids that consist of non-overlapping regular Cartesian grids. There are situations where overlapping grids have advantages, since simpler grids or substantially fewer blocks can be used. One example is the use of boundary aligned grids along the boundary and a Cartesian grid in the interior of a domain as it has been used by Linden ([7]) and Chesshire and Henshaw ([3]). Complicated grids have to be constructed without overlap. Another example is the refinement along a shock with d cells orthogonal to the shock. If rotated Cartesian overlapping grids are used, a small number of blocks (depending on the curvature) is sufficient. Using non-overlapping Cartesian grids, the number of blocks is proportional to $\frac{1}{d}$ which also introduces much overhead. Therefore we investigate the use of overlapping grids and appropriate solution methods.

THE SCHWARZ ALGORITHM ON TWO RECTANGULAR GRIDS

The Classical Schwarz Algorithm

Given a discrete problem as it arises from an elliptic partial differential equation, on two rectangular overlapping grids Ω_h^a and Ω_h^b we have the following

$$\begin{aligned} L_h u_h &= f_h \quad \text{in} \quad \Omega_h^a \cap \Omega_h^b \\ u_h &= g_h \quad \text{on} \quad \partial(\Omega_h^a \cap \Omega_h^b). \end{aligned}$$

We define the well-known Schwarz algorithm ([12]) as follows:

multiplicative (algorithm 1-m)
1. initialize u_h^a and u_h^b
2. $u_h^a \leftarrow MG(L_h^a, u_h^a, f_h^a)$ in Ω_h^a
3. $u_h^b \leftarrow I_a^b u_h^a$ on $\partial\Omega^a \cap \Omega_h^b$
4. $u_h^b \leftarrow MG(L_h^b, u_h^b, f_h^b)$ in Ω_h^b
5. $u_h^a \leftarrow I_b^a u_h^b$ on $\partial\Omega^b \cap \Omega_h^a$
6. go to step 2

additive (algorithm 1-a)
1. initialize u_h^a and u_h^b
2. $u_h^a \leftarrow MG(L_h^a, u_h^a, f_h^a)$ in Ω_h^a
3. $u_h^b \leftarrow MG(L_h^b, u_h^b, f_h^b)$ in Ω_h^b
4. $u_h^b \leftarrow I_a^b u_h^a$ on $\partial\Omega^a \cap \Omega_h^b$
5. $u_h^a \leftarrow I_b^a u_h^b$ on $\partial\Omega^b \cap \Omega_h^a$
6. go to step 2

where $MG(,,)$ is an approximative solver on a rectangular grid (we use a multilevel V(2,1) cycle). Numerical examples show that the convergence rates of both algorithms relate to each other as

$$\rho_{\text{add}} \approx \sqrt{\rho_{\text{mult}}}.$$

Thus according to the convergence behavior algorithm 1-m is two times faster than algorithm 1-a. On the other hand, in a parallel environment algorithm 1-a might turn out to be the more efficient one due to its inherent parallel potential; that is, the solution step (step 2 and 3) can be performed simultaneously. For algorithm 1-m, only parallelism in the MG solver (grid partitioning) can be exploited.

The Schwarz Algorithm as a Smoother

Similarly to algorithms 1-m and 1-a we can define a Schwarz-like relaxation scheme in the following way:

multiplicative (algorithm 2-m)
1. $u_h^a \leftarrow R^k(L_h^a, u_h^a, f_h^a)$ in Ω_h^a
2. $u_h^b \leftarrow I_a^b u_h^a$ on $\partial\Omega^a \cap \Omega_h^b$
3. $u_h^b \leftarrow R^k(L_h^b, u_h^b, f_h^b)$ in Ω_h^b
4. $u_h^a \leftarrow I_b^a u_h^b$ on $\partial\Omega^b \cap \Omega_h^a$
5. go to step 1

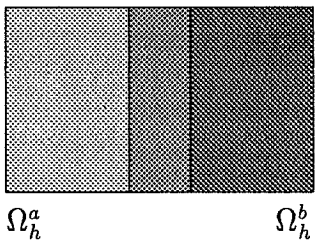
additive (algorithm 2-a)
1. $u_h^a \leftarrow R^k(L_h^a, u_h^a, f_h^a)$ in Ω_h^a
2. $u_h^b \leftarrow R^k(L_h^b, u_h^b, f_h^b)$ in Ω_h^b
3. $u_h^b \leftarrow I_a^b u_h^a$ on $\partial\Omega^a \cap \Omega_h^b$
4. $u_h^a \leftarrow I_b^a u_h^b$ on $\partial\Omega^b \cap \Omega_h^a$
5. go to step 1

where $R(,,)$ is a given relaxation scheme on the rectangular grids (e.g., Gauss Seidel).

Table 1 shows numerical results for the following test problem: Laplace's equation on a rectangular domain that consists of two overlapping grids of 65×65 points each. Here we use the standard 5 point stencil discretization, $f_h = 0$ and $g_h = 0$. For the algorithms 1-m and 1-a we use a V(2,1) multigrid cycle as an approximative solver. For the algorithms 2-m and 2-a we use $k = 1$ and a V(2,1) multigrid cycle on the whole domain. This means that we do standard coarsening on each of the grids and treat the two overlapping grids on each coarsening level as one level in the multigrid sense.

Table 1: Convergence Rates and Overlap Geometry

ovl	1-m	1-a	2-m	2-a
$2h$	0.807	0.899	0.606	0.788
$4h$	0.655	0.816	0.348	0.615
$8h$	0.432	0.667	0.154	0.384
$16h$	0.199	0.451	0.092	0.169
$32h$	0.057	0.211	0.049	0.051



For small overlap areas we observe that algorithm 2-m has the best convergence rate. Algorithm 2-a is comparable to algorithm 1-m, but due to the additivity it has a higher parallel potential. Similar results hold for other overlap geometries ([1]).

A major disadvantage of the Schwarz-like relaxation scheme is the fact that on coarser levels a bad situation of overlap occurs in a natural way. Given two fine-level overlapping grids of the

same mesh size that are aligned, the situation illustrated in figure 1 is very likely to occur. At the re-entrant corner in both the x- and the y-direction, the distance of the boundary of the grid to the closest parallel interior grid line of the other grid which goes through the interior is small compared with the mesh size. We observe a strong coupling between those two grid points that lie on an interior boundary closest to the physical boundary of the domain and, thus, a convergence rate close to 1.

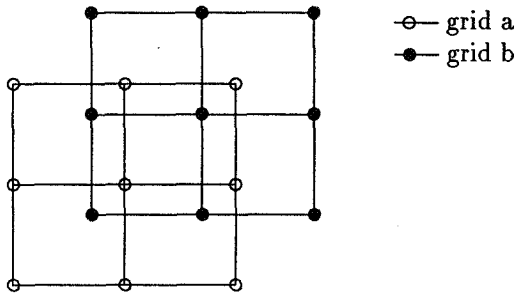


Figure 1: Bad Overlap Geometry

An Example for the Case of a Bad Overlap Geometry

The following example illustrates the situation described above. We consider the simple case of two overlapping grids of size 3×3 grid points. We discretize Poisson's equation on the union of the two grids by using the standard 5 point stencil on each of the grids. We apply the multiplicative Schwarz algorithm with an exact solver (steps 2 and 4 in algorithm 1-m). The transfer of boundary values (steps 3 and 5 of algorithm 1-m) is done by bilinear interpolation.

The spectral radius of the multiplicative Schwarz algorithm M_{Schwarz} depends on the mesh size h and the smallest distance d between the lines of the two grids:

$$\rho(M_{\text{Schwarz}}) = \frac{(h-d)^4}{h^4}.$$

For $d \searrow 0$, clearly $\rho(M_{\text{Schwarz}}) \nearrow 1$, so this example suggests that the multiplicative Schwarz algorithm is ill conditioned in the case of such overlap geometry. If we use quadratic interpolation in the x- and y-directions we observe the same behavior. Numerical experiments with various grid sizes lead to the same result.

An Approach to Improve the Convergence in the Case of a Bad Overlap Geometry

If we modify the interpolation used to transfer the interior boundaries near the physical boundary of the domain, we can overcome the bad convergence behavior that arises from a bad overlap

geometry. The bad convergence rate is due to the strong coupling between the two points that are closest to the re-entrant corner. As illustrated in figure 2 for the case of linear interpolation, we use p2 and p3 to interpolate the value on p1. Here the value on p2 is obtained by linear interpolation from the values on p4 and p5. The value on p3, in contrast to the interpolation used in the interior of the domain, is an exact boundary value, or it can be obtained by using the value on the closest grid-point on the physical boundary.

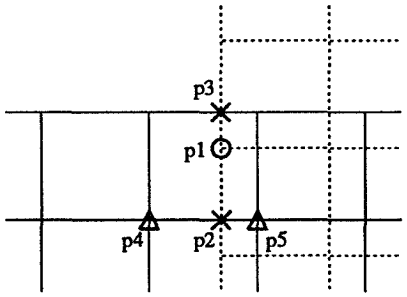


Figure 2: Interpolation of Boundary Values near a Corner

This strategy can be extended to higher order interpolation. In that case, the value on p3 has to be obtained by extrapolation from values on the physical boundary of the domain or must be given as a boundary value in the corner. Numerical examples show that, for instance in the case of linear interpolation in both the x- and y-directions, the use of the described modification of the interpolation yields a convergence rate that is smaller by a factor of ≈ 0.5 than the convergence rate with the usual interpolation.

MULTIPLICATIVE SCHWARZ ALGORITHM ON MORE THAN TWO RECTANGULAR GRIDS

In general, a refinement algorithm that produces overlapping rectangular grids will produce more than two overlapping grids. Here we show one way to get the benefits of the multiplicative Schwarz algorithm in a parallel environment. The idea is an extension of the coloring idea in relaxation schemes. In order to define a 4-step multiplicative Schwarz algorithm we have to define an overlap coloring on a family of overlapping rectangular domains $\{\Omega^i\}_{i=1\ldots k}$.

Definition 1 *(overlap coloring). We call $\phi : \{\Omega^i\} \rightarrow N$ an n-overlap coloring if for two domains Ω^i and Ω^j, with $i \neq j$ and $\Omega^i \cap \Omega^j \neq \emptyset$,*

$$\phi(\Omega^i) \neq \phi(\Omega^j)$$

and

$$\phi(\{\Omega^i\}_{i=1\ldots k}) = \{1 \ldots n\}$$

hold.

The following theorem is an application of the well-known four-color theorem.

Theorem. 1 *Let* $\bigcup_{i=1\ldots k}\Omega^i$ *be a connected set. If*

1. $\forall i,j\epsilon\{1\ldots k\}, i\neq j$: $\Omega^i-\Omega^j$ *is a connected set*
2. $\forall i,j\epsilon\{1\ldots k\}$ *with* $\Omega^i\cap\Omega^j\neq\emptyset$, *we have* $\overline{\Omega^i\cap\Omega^j}\not\subseteq\bigcup_{l\epsilon\{1\ldots k\}-\{i,j\}}\Omega^l$

then there exists a 4-overlap coloring for $\{\Omega^i\}_{i=1\ldots k}$.

Proof. First we construct a family of mutually disjoint open sets $\{\hat{\Omega}^i\}_{i=1\ldots k}$ with

$$\overline{\bigcup_{i=1}^{k}\hat{\Omega}^i}=\overline{\bigcup_{i=1}^{k}\Omega^i}.$$

This is done by defining the $\hat{\Omega}^i$ recursively as

$$\hat{\Omega}^1=\Omega^1,\qquad \hat{\Omega}^i=\Omega^i-\overline{\bigcup_{j=1}^{i-1}\Omega^j}.$$

Now we show that $\hat{\Omega}^i$ is connected for all $i\epsilon\{1\ldots k\}$. For $i\epsilon\{1,2\}$ this is trivial, for $i\geq 3$ we show this by contradiction. Suppose there is an $i\epsilon\{3\ldots k\}$ such that $\hat{\Omega}^i$ is not connected, then since $\forall j,k\epsilon\{1\ldots k\}, j\neq k$, $\Omega^j\not\subseteq\Omega^k$ and since all Ω^i are rectangular we can conclude that there is a pair of indices $j,k\neq i$ such that $\Omega^j\cap\Omega^k\subset\Omega^i$, which is a contradiction to the second hypothesis in the theorem. Since the $\hat{\Omega}^i$ are not empty we can apply the four-color theorem to obtain a coloring of the constructed domains with four colors. Now we can use this coloring for $\{\Omega^i\}_{i=1\ldots k}$ and by construction of the $\hat{\Omega}^i$ this is a 4 coloring of the Ω^i. □

This result also holds if we replace the domains by rectangular grids. So that we can obtain a 4-coloring of a family of overlapping rectangular grids and since now grids of the same color do not overlap we can solve on those nonoverlapping grids simultaneously and process the groups of equally colored grids in a multiplicative manner. Given a family of overlapping grids $\{\Omega_h^i\}_{i=1\ldots k}$ and a 4-coloring for them we can define the 4-step multiplicative Schwarz algorithm in the following way:

1. Initialize u_h^i, $i=1\ldots k$.
2. For $c=1\ldots 4$
   (a) $u_h^i\leftarrow MG(L_h^i,u_h^i,f_h^i)$ for all i such that $\Omega_h^i\epsilon\phi^{-1}(c)$.
   (b) Update the boundary points of all grids that intersect with the grids of color c.

In the context of adaptive grid refinement this algorithm yields a higher parallel potential than multiplicative processing of the grids. We have not investigated the numerical properties of such an algorithm, but the numerical results with two overlapping grids suggest that a multiplicative processing of the refinement grids has better smoothing and convergence properties than an additive algorithm. Theorem 1 provides only the existence of a 4-overlap coloring under fairly weak conditions in two dimensions. In three dimensions such a general result does not hold.

THE SCHWARZ ALGORITHM AS A SMOOTHER IN FACx

Here we want to investigate the Schwarz algorithm as a smoother in the FAC algorithm. A good reference to FAC and AFAC is [8] where a detailed description of the two algorithms is given. In [6] an implementation of the AFAC algorithm is described that is based on regular block-structured grids. We extend this idea in such a way that our implementation allows for overlap among the grids that represent one level of refinement. As a relaxation scheme we use algorithms 2-m or 2-a. In table 2 we give some numerical results for the FAC algorithm on a grid as in figure 2.

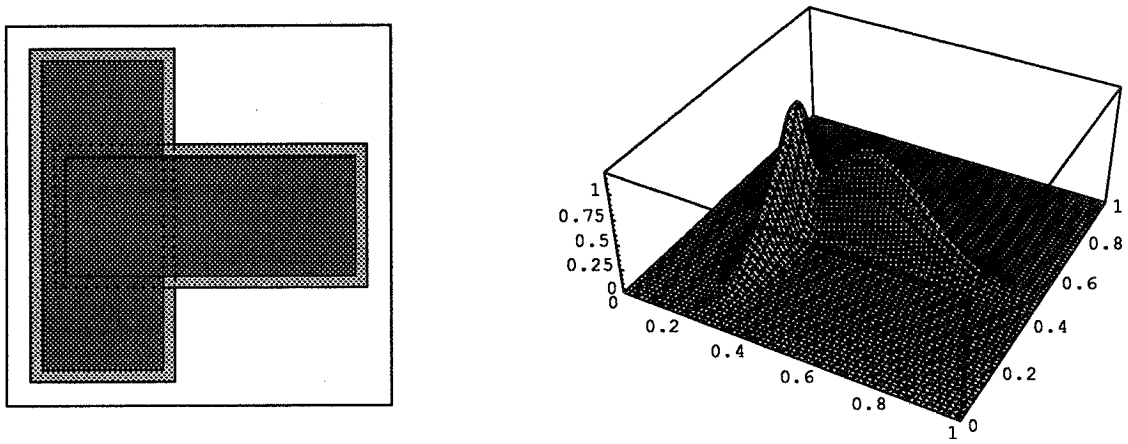


Figure 3: Grid Geometry and Exact Solution for Test Problem

Table 2: Convergence Rates for Test Problem

	additive		multiplicative	
levels	(2,1)	(4,1)	(2,1)	(4,1)
2	0.131	0.121	0.125	0.119
3	0.131	0.116	0.117	0.116
4	0.118	0.119	0.104	0.119

We solve Laplace's equation (standard 5 point discretization) on the unit square with FAC using algorithm 2-m or 2-a as a relaxation scheme on the refinement levels. Within the relaxation scheme we use Gauss-Seidel relaxation. The left picture in figure 3 shows the structure of the refinement levels, and the right picture shows a function plot of the right hand side that we used in this example. The shape of the function plot makes the region of refinement that we chose seem reasonable. In table 2 we give the average convergence rate after 10 iterations of a FACx coarse-to-fine cycle using different numbers of relaxations. We observe that there is hardly a difference between the additive and the multiplicative Schwarz relaxation. This result was already observed in the comparison of algorithm 2-m and 2-a used in a multilevel scheme. We also observe that a small number of relaxations on each refinement level is sufficient to obtain a convergence rate that is comparable to the theoretical convergence rate of FAC ([9]).

OBJECT ORIENTED IMPLEMENTATION

In our implementation we use C++ and the M++, or A++/P++ array classes (cf. [4], [10], and [11]). M++ is a commercial serial array class that provides functionality similar to Fortran 90. A++ is an array class developed by Daniel J. Quinlan at the Los Alamos National Laboratories. The user interface is compatible with M++, but in contrast to M++ the implementation focuses on the speed of the code. The first version of P++, a parallel array class that uses the SPMD programming model, was developed by Max Lemke and Daniel J. Quinlan and was based on the M++ class library. A new implementation of P++ based on A++ is currently being developed by Daniel J. Quinlan.

Due to the object oriented features of C++ and the possibility of using A++ array statements, the implementation was simplified significantly. We employed these features by dividing the FACx code into the following parts:

- A class that provides simple multilevel functionality.
- A class that provides the functionality for the Schwarz algorithm (e.g. overlap information and transfer of internal boundaries of grids).
- A class that provides the functionality needed for the FACx algorithm (e.g. transfer operators between the refinement levels).

This division made it very simple to change the code from FACx to MLAT [2] since only changes in the FACx class had to be done. This illustrates the advantages of an object oriented implementation. The code development and maintenance becomes significantly simplified.

To test the serial version of the code we used AT&T C++ and GNU C++ on a Sun SPARCstation 10. Because of the compatibility of A++ and P++ we were able to produce a parallel version of our code for the iPSC/860 within a few days. This accomplishment shows impressively the advantages of the use of an array class like A++/P++. Due to the preliminary status of the development of A++ and P++ at the time of the implementation, we were not able to obtain any interesting performance results in a parallel environment. Nevertheless, we conclude that the approach of using object oriented programming and the use of parallel array classes can be of significant use and can speed up the code development process.

CONCLUSIONS

We showed that the Schwarz algorithm can be used as a relaxation scheme in a very efficient way. It has some advantages over the approach with block structured refinement grids that consist of nonoverlapping rectangular grids. In general, one needs a smaller number of rectangular grids to cover a given domain if overlap of the grids is allowed. The fact that a larger number of

points is needed to cover a domain if overlap is allowed may be less of a disadvantage in a parallel environment. The 4-step multiplicative Schwarz algorithm further increases the existing parallel potential of the additive Schwarz algorithm if it is used on a large number of grids. The problem of bad overlap geometry does not occur when FACx or AFACx is used with regular grids and aligned refinement grids. Bad overlap geometry can be overcome by a slight modification of the interpolation that is used for the transfer of the interior boundary values in the Schwarz scheme. Finally, we can report very positive experiences with an object oriented implementation of a complex code.

REFERENCES

[1] M. Berndt. Multi-Level-Verfahren und Adaptive Verfeinerungen. Master's thesis, Universität Düsseldorf, Germany, 1994.

[2] A. Brandt. Multi–level adaptive techniques (MLAT) for partial differential equations: ideas and software. In J. R. Rice, editor, Mathematical Software III, pages 277–318. Academic Press, New York, 1977.

[3] G. Chesshire and W. D. Henshaw. Composite overlapping meshes for the solution of partial differential equations. J. Comput. Phys., 90:1–64, 1990.

[4] Dyad Corporation, Renton, OR. M++ User Manual, 1991.

[5] M. Lemke. Multilevel Verfahren mit selbst–adaptiven Gitterverfeinerungen für Parallelrechner mit verteiltem Speicher. Ph.D. thesis, Universität Düsseldorf, Germany, 1993.

[6] M. Lemke, K. Witsch, and D. Quinlan. An object–oriented approach for parallel self adaptive mesh refinement on block structured grids. In N. D. Melson, T. A. Manteuffel, and S. F. McCormick, editors, Sixth Copper Mountain Conference on Multigrid Methods, NASA CP 3224, pages 345–359, 1993.

[7] J. Linden. Mehrgitterverfahren für die Poisson–Gleichung in Kreis und Ringgebiet unter Verwendung lokaler Koordinaten. Ph.D. thesis, Institut für Angewandte Mathematik, Universität Bonn, Germany, 1981.

[8] S. F. McCormick. Multigrid Methods. Volume 3 of Frontiers in Applied Mathematics. SIAM Books, Philadelphia, 1987.

[9] S. F. McCormick and D. Quinlan. Asynchronous multilevel adaptive methods for solving partial differential equations on multiprocessors: performance results. Parallel Comput., 12:145–156, 1989.

[10] Daniel J. Quinlan. Parallel Asynchronous Fast Adaptive Composite Grid Methods. Ph.D. thesis, University of Colorado at Denver, 1993.

[11] Daniel J. Quinlan and Rebecca Parsons. A++/P++ array classes for architecture independent finite difference computations. In Proceedings of the Second Annual Object-Oriented Numerics Conference, Sunriver, Oregon, April 1994.

[12] H. A. Schwarz. Gesammelte Mathematische Abhandlungen: Volume 2. Springer-Verlag, 1980.

[13] O. B. Widlund. Some Schwarz methods for symmetric and nonsymmetric elliptic problems. In D. E. Keyes, T. F. Chan, G. Meurant, J. S. Scroggs, and R. G. Voigt, editors, Fifth International Symposium on Domain Decomposition Methods for Partial Differential Equations, pages 19–36, SIAM, Philadelphia, 1992.

FIRST-ORDER SYSTEM LEAST-SQUARES FOR THE NAVIER-STOKES EQUATIONS

P. Bochev
Department of Mathematics
Box 19408, University of Texas at Arlington
Arlington, TX 76019-0408

Z. Cai
Center for Applied Mathematical Sciences
Department of Mathematics
University of Southern California, 1042 W. 36th Place, DRB 155
Los Angeles, CA 90089-1113

T. A. Manteuffel and S. F. McCormick
Program in Applied Mathematics, Campus Box 526
University of Colorado at Boulder
Boulder, CO 80309-0526 *

SUMMARY

This paper develops a least-squares approach to the solution of the incompressible Navier-Stokes equations in primitive variables. As with our earlier work on Stokes equations, we recast the Navier-Stokes equations as a first-order system by introducing a *velocity flux* variable and associated curl and *trace* equations. We show that the resulting system is well-posed, and that an associated least-squares principle yields optimal discretization error estimates in the H^1 norm in each variable (including the velocity flux) and optimal multigrid convergence estimates for the resulting algebraic system.

INTRODUCTION

In [3], Cai, Manteuffel, and McCormick developed least-squares functionals for first-order system formulation of the Stokes equations (generalized by a pressure-perturbed form of the continuity equation to allow for linear elasticity). Full ellipticity was established of the L^2-based least-squares formulation in n dimensions by showing that the homogeneous form of

*This work was sponsored by the Air Force Office of Scientific Research under grant number AFOSR-91-0156, the National Science Foundation under grant number DMS-8704169, and the Department of Energy under grant number DE-FG03-93ER25165.

the functional is equivalent to the $(H^1)^{n^2+n+1}$ norm applied to the first-order system variables (the new n^2-component velocity flux variable, the n-component velocity variable, and the scalar pressure variable). This immediately yields optimal discretization error estimates for standard finite elements in this H^1 product norm, as well as optimal convergence estimates for multigrid methods applied to the resulting discrete systems.

The aim of this paper is to extend this methodology to the primitive variable form of the incompressible Navier-Stokes equations in two and three dimensions. We do this in the same way that the Stokes equations were reformulated based on the velocity flux variable, but now we include the nonlinear convection term in the first-order system. We recast the Euler-Lagrange equations for the least-squares principle in the canonical form $F(\lambda,\mathcal{U}) \equiv \mathcal{U}+T\cdot G(\lambda,\mathcal{U}) = 0$, where T is the least-squares solution operator for the Stokes equations. This allows us to apply conventional abstract theory and our Stokes results to obtain optimal discretization and multigrid solution estimates for each variable (including velocity flux) in the H^1 norm.

These are the first H^1 product ellipticity results for the Navier-Stokes equations that admit the practical velocity boundary conditions. Earlier work on the Stokes equations by Chang [5] used an *acceleration* variable analogous to our velocity flux; however, velocity was eliminated from the first-order system, which seems to prevent its extension to the Navier-Stokes equations, and, in any case, the formulation is limited to two dimensions. In [2], Bochev and Gunzburger developed a least-squares approach for the velocity-vorticity-pressure form of the Stokes equations, but showed that it does not allow H^1 product ellipticity in the velocity boundary condition case (a mesh weighting was introduced in the functional to obtain optimal estimates). Finally, Bochev [1] extended this methodology to the Navier-Stokes equations, but established H^1 product ellipticity only for nonstandard boundary conditions.

This paper is organized as follows: in the next section, we introduce the Navier-Stokes equations and their first-order form; in Section 3, we develop the associated least-squares principle; in Section 4, we recast this principle in canonical form and apply a corresponding abstract theory to derive error estimates; in Section 5, we establish well-posedness of the least-squares canonical form based on regularity assumptions for the original Navier-Stokes equations; and, in the final section, we develop a simple but optimal multigrid solver for the resulting discrete system. Throughout the paper we use bold face to denote vectors and underlined bold face style to denote matrices.

VELOCITY-PRESSURE-VELOCITY-FLUX NAVIER-STOKES EQUATIONS

The dimensionless equations governing the steady incompressible flow of a viscous fluid in bounded domain $\Omega \subset \mathbb{R}^n$, $n = 2, 3$, may be written in the form

$$-\nu\triangle \mathbf{u} + (\nabla \mathbf{u}^t)^t\mathbf{u} + \nabla p = \mathbf{f} \text{ in } \Omega \tag{1}$$

$$\nabla^t\mathbf{u} = 0 \text{ in } \Omega, \tag{2}$$

where $\mathbf{u}$, p, and $\mathbf{f}$ denote velocity, pressure, and given body force, respectively, and ν is the inverse of the Reynolds number, λ. The velocity variable $\mathbf{u}$ is a column vector with scalar

components u_i, so that $\nabla \mathbf{u}^t$ is a matrix with columns ∇u_i. Together with equations (1)-(2), we consider the velocity boundary condition

$$\mathbf{u} = \mathbf{0} \quad \text{on } \Gamma, \tag{3}$$

where Γ is the boundary of Ω. For uniqueness, we also impose the baseline pressure condition

$$\int_\Omega p d\Omega = 0. \tag{4}$$

To formulate the least-squares method, equations (1)-(2) will be transformed into an equivalent first-order system. The first step in this process is to introduce the *velocity flux* variable

$$\underline{\mathbf{U}} = \nabla \mathbf{u}^t, \tag{5}$$

which is a matrix with entries $U_{ij} = \partial u_j / \partial x_i$, $1 \le i, j \le n$. Then

$$(\nabla^t \underline{\mathbf{U}})^t = \triangle \mathbf{u}$$

and it is easy to see that the new variable satisfies the identities

$$\mathrm{tr}\underline{\mathbf{U}} = 0, \quad \nabla \times \underline{\mathbf{U}} = \underline{\mathbf{0}} \quad \text{in } \Omega$$

and

$$\mathbf{n} \times \underline{\mathbf{U}} = \underline{\mathbf{0}} \quad \text{on } \Gamma, \tag{6}$$

where $\mathrm{tr}\underline{\mathbf{U}} = \sum_{i=1}^n U_{ii}$ and $\mathbf{n}$ is the outward unit normal on Γ. Furthermore, the nonlinear term in (1) takes the particularly simple form

$$(\nabla \mathbf{u}^t)^t \mathbf{u} = \underline{\mathbf{U}}^t \mathbf{u}.$$

As a result, (1)-(2) can be replaced by the first-order system

$$\begin{aligned}
-\nu(\nabla^t \underline{\mathbf{U}})^t + \underline{\mathbf{U}}^t \mathbf{u} + \nabla p &= \mathbf{f} \quad \text{in } \Omega && (7)\\
\nabla^t \mathbf{u} &= 0 \quad \text{in } \Omega && (8)\\
\underline{\mathbf{U}} - \nabla \mathbf{u}^t &= \underline{\mathbf{0}} \quad \text{in } \Omega && (9)\\
\nabla(tr\underline{\mathbf{U}}) &= 0 \quad \text{in } \Omega && (10)\\
\nabla \times \underline{\mathbf{U}} &= \underline{\mathbf{0}} \quad \text{in } \Omega && (11)
\end{aligned}$$

along with conditions (3), (4), and (6).

The second step in the formulation of a suitable first-order system is to scale the momentum equation by the Reynolds number and replace the data $\mathbf{f}$ by functions with known boundary values. The resulting form of the equations will provide insight into the overall approach and facilitate error analysis of the corresponding least-squares method. For this purpose, we assume that $\mathbf{f} \in L^2(\Omega)^n$ and consider the unique solution $(\mathbf{u}_0, p_0)$ of the scaled Stokes problem

$$\begin{aligned}
-\triangle \mathbf{u} + \nabla p &= \tfrac{1}{\nu}\mathbf{f} \quad \text{in } \Omega\\
\nabla^t \mathbf{u} &= 0 \quad \text{in } \Omega\\
\mathbf{u} &= \mathbf{0} \quad \text{on } \Gamma\\
\textstyle\int_\Omega p d\Omega &= 0.
\end{aligned} \tag{12}$$

Equation (7) is then replaced by

$$-(\nabla^t \underline{\mathrm{U}})^t + \frac{1}{\nu}(\underline{\mathrm{U}} + \underline{\mathrm{U}}_0)^t(\mathrm{u} + \mathrm{u}_0) + \nabla p = 0 \text{ in } \Omega, \tag{13}$$

which determines the perturbation $(\underline{\mathrm{U}}, \mathrm{u}, \nu p)$ from the Stokes solution $(\nabla \mathrm{u}_0^t, \mathrm{u}_0^t, \nu p_0)$. To summarize, our reformulation yields the system

$$\begin{aligned} -(\nabla^t \underline{\mathrm{U}})^t + \frac{1}{\nu}(\underline{\mathrm{U}} + \underline{\mathrm{U}}_0)^t(\mathrm{u} + \mathrm{u}_0) + \nabla p &= 0 \text{ in } \Omega && (14)\\ \nabla^t \mathrm{u} &= 0 \text{ in } \Omega && (15)\\ \underline{\mathrm{U}} - \nabla \mathrm{u}^t &= \underline{0} \text{ in } \Omega && (16)\\ \nabla(tr\underline{\mathrm{U}}) &= 0 \text{ in } \Omega && (17)\\ \nabla \times \underline{\mathrm{U}} &= \underline{0} \text{ in } \Omega. && (18) \end{aligned}$$

LEAST-SQUARES METHOD

The least-squares functional for first-order system (14)-(18), (3), (4), and (6) is defined as follows:

$$\begin{aligned} J(\underline{\mathrm{U}}, \mathrm{u}, p) &= \| -(\nabla^t \underline{\mathrm{U}})^t + \frac{1}{\nu}(\underline{\mathrm{U}} + \underline{\mathrm{U}}_0)^t(\mathrm{u} + \mathrm{u}_0) + \nabla p\|_0^2 \\ &+ \|\nabla^t \mathrm{u}\|_0^2 + \|\underline{\mathrm{U}} - \nabla \mathrm{u}^t\|_0^2 + \|\nabla(tr\underline{\mathrm{U}})\|_0^2 + \|\nabla \times \underline{\mathrm{U}}\|_0^2. \end{aligned} \tag{19}$$

Note that our scaling of (7) by the Reynolds number is equivalent to the use of an L^2 norm weighted by λ for the residual of this equation; see also [3].

To define the least-squares method, we need a suitable minimization problem. Let

$$\mathbf{X} = \{(\underline{\mathrm{U}}, \mathrm{u}, p) \in H^1(\Omega)^{n^2} \times H^1(\Omega)^n \times H^1(\Omega) \cap L_0^2(\Omega) \,\big|\, \mathrm{u} = 0, \mathrm{n} \times \underline{\mathrm{U}} = \underline{0} \text{ on } \Gamma\}, \tag{20}$$

where $L_0^2(\Omega) = \{p \in L^2(\Omega) \big| \int_\Omega p d\Omega = 0\}$. Then the least-squares principle for functional (19) is

Find $(\underline{\mathrm{U}}, \mathrm{u}, p) \in \mathbf{X}$ such that

$$J(\underline{\mathrm{U}}, \mathrm{u}, p) \leq J(\underline{\mathrm{V}}, \mathrm{v}, q) \quad \text{for all } (\underline{\mathrm{V}}, \mathrm{v}, q) \in \mathbf{X}. \tag{21}$$

It is easy to see that the Euler-Lagrange equation for this minimization problem is given by the variational problem

Find $(\underline{\mathrm{U}}, \mathrm{u}, p) \in \mathbf{X}$ such that

$$\mathcal{B}((\underline{\mathbf{U}},\mathbf{u},p),(\underline{\mathbf{V}},\mathbf{v},p)) \equiv$$
$$\Big(-(\nabla^t\underline{\mathbf{U}})^t + \frac{1}{\nu}(\underline{\mathbf{U}}+\underline{\mathbf{U}}_0)^t(\mathbf{u}+\mathbf{u}_0) + \nabla p,$$
$$-(\nabla^t\underline{\mathbf{V}})^t + \frac{1}{\nu}\left((\underline{\mathbf{U}}+\underline{\mathbf{U}}_0)^t\mathbf{v} + \underline{\mathbf{V}}^t(\mathbf{u}+\mathbf{u}_0)\right) + \nabla q\Big)_0 +$$
$$\left(\nabla^t\mathbf{u},\nabla^t\mathbf{v}\right)_0 + (\nabla(tr\underline{\mathbf{U}}),\nabla(tr\underline{\mathbf{V}}))_0 +$$
$$\left(\underline{\mathbf{U}}-\nabla\mathbf{u}^t,\underline{\mathbf{V}}-\nabla\mathbf{v}^t\right)_0 + (\nabla\times\underline{\mathbf{U}},\nabla\times\underline{\mathbf{V}})_0 = 0 \tag{22}$$

for all $(\underline{\mathbf{V}},\mathbf{v},q)\in\mathbf{X}$.

Let $\mathbf{X}_h$ denote a finite-dimensional subspace of $\mathbf{X}$. Then the least-squares discretization method for the Navier-Stokes equations is defined by the following discrete variational problem:

Find $(\underline{\mathbf{U}}^h,\mathbf{u}^h,p^h)\in\mathbf{X}_h$ such that

$$\mathcal{B}((\underline{\mathbf{U}}^h,\mathbf{u}^h,p^h),(\underline{\mathbf{V}}^h,\mathbf{v}^h,p^h)) = 0 \quad \text{for all } (\underline{\mathbf{V}}^h,\mathbf{v}^h,q^h)\in\mathbf{X}_h. \tag{23}$$

It is easy to see that the discrete variational problem (23) corresponds to the necessary condition for the following discrete least-squares principle for (19):

Find $(\underline{\mathbf{U}}^h,\mathbf{u}^h,p^h)\in\mathbf{X}_h$ such that

$$J(\underline{\mathbf{U}}^h,\mathbf{u}^h,p^h) \le J(\underline{\mathbf{V}}^h,\mathbf{v}^h,q^h) \quad \text{for all } (\underline{\mathbf{V}}^h,\mathbf{v}^h,q^h)\in\mathbf{X}_h. \tag{24}$$

For the space $\mathbf{X}_h$, we assume the following approximation property: there exists an integer $d\ge 0$ such that, for all $\underline{\mathbf{U}}\in H^{d+1}(\Omega)^{n^2}$, $\mathbf{u}\in H^{d+1}(\Omega)^n$, and $p\in H^{d+1}(\Omega)$, one can find $(\underline{\mathbf{U}}^h,\mathbf{u}^h,p^h)\in\mathbf{X}_h$ with

$$\|\underline{\mathbf{U}}-\underline{\mathbf{U}}^h\|_r + \|\mathbf{u}-\mathbf{u}^h\|_r + \|p-p^h\|_r \le Ch^{d+1-r}\left(\|\underline{\mathbf{U}}\|_{d+1} + \|\mathbf{u}\|_{d+1} + \|p\|_{d+1}\right), \tag{25}$$

$r=0,1$.

DISCRETIZATION ERROR ESTIMATES

The main goal of this section is to derive error estimates for least-squares method (23). For this purpose, we show how to cast nonlinear problems (22) and (23) in the respective canonical forms

$$F(\lambda,\mathcal{U}) \equiv \mathcal{U} + T\cdot G(\lambda,\mathcal{U}) = 0 \tag{26}$$

and

$$F^h(\lambda,\mathcal{U}^h) \equiv \mathcal{U}^h + T_h\cdot G(\lambda,\mathcal{U}^h) = 0\,. \tag{27}$$

This will allow us to apply the abstract approximation theory of [6]. The following function spaces will be needed in the sequel (with some nonnegative integer m):

$$\mathbf{X}^m = \left[H^{m+1}(\Omega)^{n^2} \times H^{m+1}(\Omega)^n \times H^{m+1}(\Omega)\right] \cap \mathbf{X}, \tag{28}$$

$$\mathbf{Y} = \mathbf{X}^*, \tag{29}$$

$$\mathbf{Z} = L^{3/2}(\Omega)^{n^2} \times L^{3/2}(\Omega)^n \times L^{3/2}(\Omega); \tag{30}$$

where $\mathbf{X}^*$ denotes the dual of $\mathbf{X}$ with respect to the L^2 inner product. The approximation in (27) is introduced by way of the operator T_h. Therefore, the error estimates will depend largely on the nature of the operator T and its approximation T_h. The basic idea is to define T to be the least-squares Stokes solution operator and T_h to be its finite element approximation. The approximation properties of these choices have been studied in [3]. Now, once T is known, the operator G is then defined by the remaining terms in (22). The key is that the corresponding nonlinear part for T_h is also G, as we assert in our first lemma.

With this in mind, we make the identifications $\mathcal{U} = (\underline{\mathbf{U}}, \mathbf{u}, p)$, $\mathcal{U}^h = (\underline{\mathbf{U}}^h, \mathbf{u}^h, p^h)$, $\mathcal{V} = (\underline{\mathbf{V}}, \mathbf{v}, q)$, $\mathcal{V}^h = (\underline{\mathbf{V}}^h, \mathbf{v}^h, q^h)$, and $\lambda = 1/\nu$, and we assume that $\lambda \in \Lambda$, where Λ is a compact subset of $\mathbf{R}^+$. We then introduce the following:

$T : \mathbf{Y} \mapsto \mathbf{X}$ defined by $\mathcal{U} = T\mathbf{g}$ for $\mathbf{g} \in \mathbf{Y}$ if and only if

$$\begin{aligned} \mathcal{B}_S(\mathcal{U}, \mathcal{V}) \;\equiv\; & \left(-(\nabla^t \underline{\mathbf{U}})^t + \nabla p, -(\nabla^t \underline{\mathbf{V}})^t + \nabla q\right)_0 \\ + \;& \left(\nabla^t \mathbf{u}, \nabla^t \mathbf{v}\right)_0 + (\nabla(tr\,\underline{\mathbf{U}}), \nabla(tr\,\underline{\mathbf{V}}))_0 \\ + \;& \left(\underline{\mathbf{U}} - \nabla \mathbf{u}^t, \underline{\mathbf{V}} - \nabla \mathbf{v}^t\right)_0 + (\nabla \times \underline{\mathbf{U}}, \nabla \times \underline{\mathbf{V}})_0 \\ = \;& (\mathbf{g}_1, \underline{\mathbf{V}}) + (\mathbf{g}_2, \mathbf{v}) + (\mathbf{g}_3, q) \end{aligned} \tag{31}$$

for all $(\underline{\mathbf{V}}, \mathbf{v}, q) \in \mathbf{X}$;

$T_h : \mathbf{Y} \mapsto \mathbf{X}_h$ defined by $\mathcal{U}^h = T\mathbf{g}$ for $\mathbf{g} \in \mathbf{Y}$ if and only if

$$\mathcal{B}_S(\mathcal{U}^h, \mathcal{V}^h) = (\mathbf{g}_1, \underline{\mathbf{V}}^h) + (\mathbf{g}_2, \mathbf{v}^h) + (\mathbf{g}_3, q^h) \quad \text{for all } (\underline{\mathbf{V}}^h, \mathbf{v}^h, q^h) \in \mathbf{X}_h; \tag{32}$$

and

$G : \Lambda \times \mathbf{X} \to \mathbf{Y}$ with $\mathbf{g} = G(\lambda, \mathcal{U})$ for $\mathcal{U} \in \mathbf{X}$ if and only if

$$\begin{aligned} &(\mathbf{g}_1, \underline{\mathbf{V}}) + (\mathbf{g}_2, \mathbf{v}) + (\mathbf{g}_3, q) = \\ &\quad \left(-(\nabla^t \underline{\mathbf{U}})^t + \nabla p, \frac{1}{\nu}\left((\underline{\mathbf{U}} + \underline{\mathbf{U}}_0)^t \mathbf{v} + \underline{\mathbf{V}}^t(\mathbf{u} + \mathbf{u}_0)\right)\right)_0 + \\ &\quad \left(\frac{1}{\nu}(\underline{\mathbf{U}} + \underline{\mathbf{U}}_0)^t(\mathbf{u} + \mathbf{u}_0)\,, \right. \\ &\qquad \left. -(\nabla^t \underline{\mathbf{V}})^t + \nabla q + \frac{1}{\nu}\left((\underline{\mathbf{U}} + \underline{\mathbf{U}}_0)^t \mathbf{v} + \underline{\mathbf{V}}^t(\mathbf{u} + \mathbf{u}_0)\right)\right)_0 \end{aligned} \tag{33}$$

for all $(\underline{\mathrm{V}}, \mathrm{v}, q) \in \mathbf{X}$.

We then have the following equivalence.

Lemma 1. *Assume that T, T_h, and G are defined by (31), (32), and (33), respectively. Then nonlinear problem (22) is equivalent to (26) and discrete nonlinear problem (23) is equivalent to (27).*

Proof. Assume that $\mathcal{U} = (\underline{\mathrm{U}}, \mathrm{u}, p)$ solves problem (26) with T and G given by (31) and (33), respectively. Then $\mathcal{U} = -T\mathrm{g}$ if and only if

$$\mathcal{B}_S(\mathcal{U}, \mathcal{V}) = (\mathrm{g}, \mathcal{V}) \quad \text{for all} \quad \mathcal{V} \in \mathbf{X},$$

and $\mathrm{g} = G(\lambda, \mathcal{U})$ if and only if (33) holds. It follows that $\mathcal{U}$ also solves variational problem (22). Conversely, if $\mathcal{U}$ solves (22), let g be defined by (33). Then $\mathcal{B}_S(\mathcal{U}, \mathcal{V}) = (\mathrm{g}, \mathcal{V})$ for all $\mathcal{V} \in \mathbf{X}$, i.e., $\mathcal{U} = -T\mathrm{g}$. Thus, (22) and (26) are equivalent. Proof of the equivalence of (23) and (27) is identical. □

The error estimates for the least-squares method (23) can now be derived from the abstract approximation theory of [6]. Below we state the main result of this theory in terms of general T and T_h but specialized to our needs. Here we let $D_{\mathcal{U}}G(\lambda, \mathcal{U})$ and $D_{\mathcal{U}}F(\lambda, \mathcal{U})$ denote the Fréchet derivative of G and F with respect to $\mathcal{U}$.

Theorem 1. *Assume that $\{(\lambda, \mathcal{U}(\lambda)) \mid \lambda \in \Lambda\}$ is a branch of regular solutions of (26), i.e., that $\lambda \mapsto \mathcal{U}(\lambda)$ is a continuous map $\Lambda \mapsto \mathbf{X}$ and that $D_{\mathcal{U}}F(\lambda, \mathcal{U})$ is an isomorphism of $\mathbf{X}$, where $F(\lambda, \mathcal{U}) = 0$ is abstract form (26). Furthermore, assume that $T \in L(\mathbf{Y}, \mathbf{X})$ and that G is a C^2 map $\Lambda \times \mathbf{X} \mapsto \mathbf{Y}$, such that all second derivatives of G are bounded on bounded subsets of $\Lambda \times \mathbf{X}$. Finally, assume that there exists a space $\mathbf{Z} \subset \mathbf{Y}$, with continuous imbedding, such that $D_{\mathcal{U}}G(\lambda, \mathcal{U}) \in L(\mathbf{X}, \mathbf{Z})$ for all $\lambda \in \Lambda$ and $\mathcal{U} \in \mathbf{X}$. If approximate problem (27) is such that*

$$\lim_{h \to 0} \|(T - T_h)\mathrm{g}\|_{\mathbf{X}} = 0 \tag{34}$$

for all $\mathrm{g} \in \mathbf{Y}$ and

$$\lim_{h \to 0} \|T - T_h\|_{L(\mathbf{Z}, \mathbf{X})} = 0\,, \tag{35}$$

then:

1. *there exists a neighborhood $\mathcal{O}$ of the origin in $\mathbf{X}$ and, for h sufficiently small, a unique C^2 function $\lambda \mapsto \mathcal{U}^h(\lambda) \in \mathbf{X}_h$ such that $\{(\lambda, \mathcal{U}^h(\lambda)) \mid \lambda \in \Lambda\}$ is a branch of regular solutions of the discrete problem (27) and $\mathcal{U}(\lambda) - \mathcal{U}^h(\lambda) \in \mathcal{O}$ for all $\lambda \in \Lambda$;*

2. *there exists a positive constant C, independent of h and λ, such that*

$$\|\mathcal{U}^h(\lambda) - \mathcal{U}(\lambda)\|_{\mathbf{X}} \leq C\|(T - T^h)G(\lambda, \mathcal{U}(\lambda))\|_{\mathbf{X}} \tag{36}$$

uniformly in λ;

3. *if the regular branch is such that* $\mathcal{U}(\lambda) \in \mathbf{X}^m$ *for some integer* $m \geq 1$ *and* $\tilde{d} \equiv \min\{d, m\}$, *where* d *is the integer from (25), then*

$$\begin{aligned} &\|\underline{\mathbf{U}}(\lambda) - \underline{\mathbf{U}}^h(\lambda)\|_1 + \|\mathbf{u}(\lambda) - \mathbf{u}^h(\lambda)\|_1 + \|p(\lambda) - p^h(\lambda)\|_1 \\ &\quad \leq \; Ch^{\tilde{d}} \left(\|\underline{\mathbf{U}}(\lambda)\|_{\tilde{d}+1} + \|\mathbf{u}(\lambda)\|_{\tilde{d}+1} + \|p(\lambda)\|_{\tilde{d}+1} \right). \end{aligned} \tag{37}$$

In the next few lemmas, we verify the hypotheses of Theorem 1 for our least-squares formulation. We begin by establishing essential properties of the operators T and T_h, which we henceforth assume are defined by (31) and (32), respectively.

Lemma 2. $T \in L(\mathbf{Y}, \mathbf{X})$ *and* $T_h \in L(\mathbf{Y}, \mathbf{X}_h)$.

Proof. The form $\mathcal{B}_S(\cdot, \cdot)$ is continuous and coercive on $\mathbf{X} \times \mathbf{X}$ (see [3]) and, by virtue of the inclusion $\mathbf{X}_h \subset \mathbf{X}$, it is also continuous and coercive on $\mathbf{X}_h \times \mathbf{X}_h$. Furthermore, $(\mathbf{g}, \mathcal{V})$ defines a continuous functional on $\mathbf{X} \ni \mathcal{V} \mapsto \mathbf{R}$ for each $\mathbf{g} \in \mathbf{Y}$. Thus, the Lax-Milgram Theorem implies that, for all $\mathbf{g} \in \mathbf{Y}$, variational problems (31) and (32) have unique respective solutions $\mathcal{U} \in \mathbf{X}$ and $\mathcal{U}^h \in \mathbf{X}_h$, i.e., $T : \mathbf{Y} \mapsto \mathbf{X}$ and $T_h : \mathbf{Y} \mapsto \mathbf{X}_h$ are well defined linear operators. From

$$C\|\mathcal{U}\|_{\mathbf{X}}^2 \leq \mathcal{B}_S(\mathcal{U}, \mathcal{U}) = (\mathbf{g}, \mathcal{U}) \leq \|\mathbf{g}\|_{\mathbf{Y}} \|\mathcal{U}\|_{\mathbf{X}},$$

it follows that

$$\|T\mathbf{g}\|_{\mathbf{X}} = \|\mathcal{U}\|_{\mathbf{X}} \leq C\|\mathbf{g}\|_{\mathbf{Y}},$$

i.e., T is in $L(\mathbf{Y}, \mathbf{X})$. The proof that $T_h \in L(\mathbf{Y}, \mathbf{X}_h)$ is similar. □

Before continuing with the approximation properties of T_h, consider the choice of $\mathbf{Y}$ and $\mathbf{Z}$ in (29) and (30). When $\mathbf{Z} \subset \mathbf{Y}$ with compact imbedding, the proof of (35) can be simplified. First, note that $\mathbf{Y}$ is not identical to a product of $H^{-1}(\Omega)$ spaces. For instance, with $\underline{\mathbf{U}}_i$ denoting the ith column of $\underline{\mathbf{U}}$, then $\underline{\mathbf{U}}_i \in \mathbf{H}_t^1(\Omega) = \{\mathbf{v} \in H^1(\Omega)^n \,|\mathbf{n} \times \mathbf{v} = 0 \quad \text{on } \Gamma\}$, whose dual is not $H^{-1}(\Omega)^n$. As a result, $\mathbf{Z}$ will be compactly imbedded in $\mathbf{Y}$ if $L^{3/2}(\Omega)$ is compactly imbedded in the duals of $H_0^1(\Omega)$, $\mathbf{H}_t^1(\Omega)$, and $H^1(\Omega)$. The first imbedding follows from Sobolev's Imbedding Theorem; see, e.g., [6]. Compactness of the other two imbeddings can be shown along the following lines. Since components of $\mathbf{H}_t^1(\Omega)$ and the space $H^1(\Omega)$ are compactly imbedded in $L^3(\Omega)$ and the adjoint of a compact operator is compact, it follows that $L^{3/2}(\Omega)^n$ and $L^{3/2}(\Omega)$ are imbedded compactly in the dual spaces of $\mathbf{H}_t^1(\Omega)$ and $H^1(\Omega)$.

Lemma 3. *Convergence properties (34) and (35) hold. If, in addition,* $\mathbf{g} \in \mathbf{Y}$ *is such that* $T\mathbf{g} \in \mathbf{X}^m$ *for some* $m \geq 1$ *and* $\tilde{d} = \min(d, m)$, *where* d *is the integer from (25), then*

$$\|(T - T_h)\mathbf{g}\|_{\mathbf{X}} \leq Ch^{\tilde{d}} \|T\mathbf{g}\|_{\mathbf{X}^{\tilde{d}+1}}. \tag{38}$$

Proof. First note that (35) follows from (34) when the imbedding $\mathbf{Z} \subset \mathbf{Y}$ is compact. It thus suffices to establish (34); that is,

$$\|(T - T_h)\mathbf{g}\|_{\mathbf{X}} = \|\underline{\mathbf{U}} - \underline{\mathbf{U}}^h\|_1 + \|\mathbf{u} - \mathbf{u}^h\|_1 + \|p - p^h\|_1 \to 0$$

when $h \to 0$. Recall that $T : \mathbf{Y} \mapsto \mathbf{X}$. Therefore, from $\mathbf{g} \in \mathbf{Y}$ it follows that $\mathcal{U} \in \mathbf{X}$; that is, $\underline{\mathbf{U}} \in H^1(\Omega)^{n^2}$, $\mathbf{u} \in H^1(\Omega)^n$, and $p \in H^1(\Omega)$. Then the above limit follows from Cea's Lemma and the standard approximation result for $v \in H^1(\Omega)$:

$$\lim_{h \to 0} \inf_{v^h} \|v - v^h\|_1 = 0 .$$

(See [4] for an analogous result for scalar elliptic equations.)

To prove the second part of the lemma, suppose $\mathcal{U} = T\mathbf{g} \in \mathbf{X}^m$. Then an immediate consequence of the continuity and coercivity of $\mathcal{B}_S(\cdot,\cdot)$ is the Stokes error estimate

$$\|(T-T_h)\mathbf{g}\|_{\mathbf{X}} = \|\underline{\mathbf{U}}-\underline{\mathbf{U}}^h\|_1+\|\mathbf{u}-\mathbf{u}^h\|_1+\|p-p^h\|_1 \le Ch^{\tilde{d}} \left(\|\underline{\mathbf{U}}\|_{\tilde{d}+1} + \|\mathbf{u}\|_{\tilde{d}+1} + \|p\|_{\tilde{d}+1}\right) .$$

□

The only hypotheses of Theorem 1 that remain to be verified are the assumptions concerning the nonlinear operator G. For this purpose, we need the weak and strong forms of the first Fréchet derivative $D_{\mathcal{U}}G(\lambda,\mathcal{U})$ and the weak form of the second Fréchet derivative $D^2_{\mathcal{U}}G(\lambda,\mathcal{U})$. To determine the weak form of $D_{\mathcal{U}}G(\lambda,\mathcal{U})$, let $\hat{\mathcal{U}} \in \mathbf{X}$, substitute $\mathcal{U}+\hat{\mathcal{U}}$ into (33), and expand about $\mathcal{U}$. This yields the following weak representation of $D_{\mathcal{U}}G(\lambda,\mathcal{U})$:

> $D_{\mathcal{U}}G(\lambda,\mathcal{U}) : \Lambda \times \mathbf{X} \to \mathbf{Y}$ defined by $\mathbf{g} = D_{\mathcal{U}}G(\lambda,\mathcal{U})\hat{\mathcal{U}}$ for $\mathcal{U} \in \mathbf{X}$ if and only if

$$\begin{aligned}
(\mathbf{g}_1, \underline{\mathbf{V}}) + (\mathbf{g}_2, \mathbf{v}) + (\mathbf{g}_3, q) = \\
&\left(-(\nabla^t \underline{\mathbf{U}})^t + \nabla p, \frac{1}{\nu}\left(\hat{\underline{\mathbf{U}}}^t \mathbf{v} + \underline{\mathbf{V}}^t \hat{\mathbf{u}}\right)\right)_0 + \\
&\left(-(\nabla^t \hat{\underline{\mathbf{U}}})^t + \nabla \hat{p}, \frac{1}{\nu}\left((\underline{\mathbf{U}} + \underline{\mathbf{U}}_0)^t \mathbf{v} + \underline{\mathbf{V}}^t(\mathbf{u}+\mathbf{u}_0)\right)\right)_0 + \\
&\left(\frac{1}{\nu}(\underline{\mathbf{U}} + \underline{\mathbf{U}}_0)^t(\mathbf{u}+\mathbf{u}_0), \frac{1}{\nu}\left(\hat{\underline{\mathbf{U}}}^t \mathbf{v} + \underline{\mathbf{V}}^t \hat{\mathbf{u}}\right)\right)_0 + \\
&\left(\frac{1}{\nu}\left((\underline{\mathbf{U}} + \underline{\mathbf{U}}_0)^t \hat{\mathbf{u}} + \hat{\underline{\mathbf{U}}}^t(\mathbf{u}+\mathbf{u}_0)\right),\right. \\
&\qquad \left. -(\nabla^t \underline{\mathbf{V}})^t + \nabla q + \frac{1}{\nu}\left((\underline{\mathbf{U}} + \underline{\mathbf{U}}_0)^t \mathbf{v} + \underline{\mathbf{V}}^t(\mathbf{u}+\mathbf{u}_0)\right)\right)_0
\end{aligned} \tag{39}$$

> for all $(\underline{\mathbf{V}}, \mathbf{v}, q) \in \mathbf{X}$.

The strong form of $D_{\mathcal{U}}G(\lambda,\mathcal{U})\hat{\mathcal{U}}$ can be found from (39) using standard integration by parts:

$$\begin{aligned}
\mathbf{g}_1 &= \frac{1}{\nu}\hat{\mathbf{u}}\left(-(\nabla^t \underline{\mathbf{U}})^t + \nabla p + \frac{1}{\nu}(\underline{\mathbf{U}} + \underline{\mathbf{U}}_0)^t(\mathbf{u}+\mathbf{u}_0)\right)^t \\
&+ \frac{1}{\nu}(\mathbf{u}+\mathbf{u}_0)\left(-(\nabla^t \hat{\underline{\mathbf{U}}})^t + \nabla \hat{p} + \frac{1}{\nu}\left((\underline{\mathbf{U}} + \underline{\mathbf{U}}_0)^t \hat{\mathbf{u}} + \hat{\underline{\mathbf{U}}}^t(\mathbf{u}+\mathbf{u}_0)\right)\right)^t \\
&+ \frac{1}{\nu}\nabla\left((\underline{\mathbf{U}} + \underline{\mathbf{U}}_0)^t \hat{\mathbf{u}} + \hat{\underline{\mathbf{U}}}^t(\mathbf{u}+\mathbf{u}_0)\right)^t ,
\end{aligned} \tag{40}$$

$$\mathbf{g}_2 = \frac{1}{\nu}\hat{\underline{\mathbf{U}}}\left(-(\nabla^t\underline{\mathbf{U}})^t + \nabla p + \frac{1}{\nu}(\underline{\mathbf{U}}+\underline{\mathbf{U}}_0)^t(\mathbf{u}+\mathbf{u}_0)\right)$$
$$+ \frac{1}{\nu}(\underline{\mathbf{U}}+\underline{\mathbf{U}}_0)\left(-(\nabla^t\hat{\underline{\mathbf{U}}})^t + \nabla\hat{p} + \frac{1}{\nu}\left((\underline{\mathbf{U}}+\underline{\mathbf{U}}_0)^t\hat{\mathbf{u}} + \hat{\underline{\mathbf{U}}}^t(\mathbf{u}+\mathbf{u}_0)\right)\right), \quad (41)$$

$$\mathbf{g}_3 = -\frac{1}{\nu}\nabla^t\left((\underline{\mathbf{U}}+\underline{\mathbf{U}}_0)^t\hat{\mathbf{u}} + \hat{\underline{\mathbf{U}}}^t(\mathbf{u}+\mathbf{u}_0)\right), \quad (42)$$

for all $(\underline{\mathbf{V}}, \mathbf{v}, q) \in \mathbf{X}$.

Finally, the weak form of the second Fréchet derivative is

$D^2_{\mathcal{U}}G(\lambda,\mathcal{U}) : \Lambda \times [\mathbf{X}\times\mathbf{X}] \to \mathbf{Y}$ defined by $\mathbf{g} = D^2_{\mathcal{U}}G(\lambda,\mathcal{U})[\hat{\mathcal{U}},\hat{\hat{\mathcal{U}}}]$ for $\mathcal{U} \in \mathbf{X}$ if and only if

$$(\mathbf{g}_1,\underline{\mathbf{U}}) + (\mathbf{g}_2,\mathbf{v}) + (\mathbf{g}_3,q) =$$
$$\left(-(\nabla^t\hat{\underline{\mathbf{U}}})^t + \nabla\hat{\hat{p}} + \frac{1}{\nu}\left(\hat{\hat{\underline{\mathbf{U}}}}^t(\mathbf{u}+\mathbf{u}_0) + (\underline{\mathbf{U}}+\underline{\mathbf{U}}_0)^t\hat{\mathbf{u}}\right),\right.$$
$$\left.\frac{1}{\nu}\left(\hat{\underline{\mathbf{U}}}^t\mathbf{v} + \underline{\mathbf{V}}^t\hat{\mathbf{u}}\right)\right)_0 +$$
$$\frac{1}{\nu}\left(-(\nabla^t\hat{\underline{\mathbf{U}}})^t + \nabla\hat{p} + \hat{\underline{\mathbf{U}}}^t(\mathbf{u}+\mathbf{u}_0) + (\underline{\mathbf{U}}+\underline{\mathbf{U}}_0)^t\hat{\mathbf{u}}, \frac{1}{\nu}(\hat{\hat{\underline{\mathbf{U}}}}^t\mathbf{v} + \underline{\mathbf{V}}^t\hat{\hat{\mathbf{u}}})\right)_0 +$$
$$\left(\frac{1}{\nu}\left(\hat{\underline{\mathbf{U}}}^t\hat{\mathbf{u}} + \hat{\hat{\underline{\mathbf{U}}}}^t\hat{\mathbf{u}}\right),\right.$$
$$\left.-(\nabla^t\underline{\mathbf{V}})^t + \nabla q + \frac{1}{\nu}\left((\underline{\mathbf{U}}+\underline{\mathbf{U}}_0)^t\mathbf{u} + \underline{\mathbf{V}}^t(\mathbf{u}+\mathbf{u}_0)\right)\right)_0 \quad (43)$$

for all $(\underline{\mathbf{V}}, \mathbf{v}, q) \in \mathbf{X}$.

The next lemma summarizes the technical results that we use in the sequel.

Lemma 4. *Let D_i denote the derivative with respect to the i^{th} coordinate variable in $\mathbb{R}^n$, $1 \le i \le n$, and assume that u, v, w, and z are in $H^1(\Omega)$. Then*

$$\left|\int_\Omega D_i u\, v\, w\, d\Omega\right| \le C\|u\|_1\,\|v\|_1\,\|w\|_1, \quad (44)$$

$1 \le i \le n$, *and*

$$\left|\int_\Omega u\, v\, w\, z\, d\Omega\right| \le C\|u\|_1\,\|v\|_1\,\|w\|_1\,\|z\|_1. \quad (45)$$

The mapping $(u,v) \mapsto uv$ is a continuous bilinear mapping from $L^2(\Omega)\times H^1(\Omega)$ into $L^{3/2}(\Omega)$ and the mapping $(u,v,w) \mapsto uvw$ is a continuous trilinear mapping from $H^1(\Omega)\times H^1(\Omega)\times H^1(\Omega)$ into $L^{3/2}(\Omega)$. That is,

$$\|uv\|_{0,3/2} \le C\|u\|_{0,2}\|v\|_{1,2} \quad \textit{for all } u \in L^2(\Omega) \textit{ and } v \in H^1(\Omega), \quad (46)$$

$$\|uvw\|_{0,3/2} \le C\|u\|_{1,2}\|v\|_{1,2}\|w\|_{1,2} \quad \textit{for all } u,v,w \in H^1(\Omega)\,. \tag{47}$$

Proof. The first part of the lemma follows easily from the imbedding $H^1(\Omega) \subset L^4(\Omega)$ in two and three dimensions and the Hölder inequality. The second part follows directly from a result in [6]. □

For a more general version of (46) and (47), see [7].

In the next lemma, we establish properties of G that are required for the validity of the approximation result in Theorem 1.

Lemma 5. *Assume that the mapping G is defined by (33). For* $\mathbf{X}$, $\mathbf{Y}$, *and* $\mathbf{Z}$ *given by (20), (29), and (30), respectively, the following are true:*

1. *For all* $\mathcal{U} \in \mathbf{X}$, $D_{\mathcal{U}}G(\lambda,\mathcal{U}) \in L(\mathbf{X},\mathbf{Z})$.
2. *The second Fréchet derivative* $D^2_{\mathcal{U}}G(\lambda,\mathcal{U})$ *is bounded on bounded subsets of* $\Lambda \times \mathbf{X}$.

Proof. To prove 1, consider strong form (40)-(42) of $D_{\mathcal{U}}G(\lambda,\mathcal{U})$. By assumption, $\mathcal{U} \in \mathbf{X}$; that is, $\underline{\mathrm{U}} \in H^1(\Omega)^{n^2}$, $\mathrm{u} \in H^1(\Omega)^n$, and $p \in H^1(\Omega)$. Now each of the equations (40), (41), and (42) consists of terms of the form $D_i u\, v$ and uvw, where u, v, and w belong to $H^1(\Omega)$, so the second part of Lemma 4 implies that $(\mathrm{g}_1, \mathrm{g}_2, \mathrm{g}_3) \in \mathbf{Z}$. Using (46) and (47), it also follows that

$$\|D_{\mathcal{U}}G(\lambda,\mathcal{U})\hat{\mathcal{U}}\|_{\mathbf{Z}} \le C\|\hat{\mathcal{U}}\|_{\mathbf{X}}\,,$$

i.e., that $D_{\mathcal{U}}G(\lambda,\mathcal{U}) \in L(\mathbf{X},\mathbf{Z})$.

To prove 2, consider weak form (43) of the second Fréchet derivative. Assume that $(\lambda,\mathcal{U})$ belongs to a bounded subset of $\Lambda \times \mathbf{X}$ and let $\hat{\mathcal{U}} \in \mathbf{X}$, and $\hat{\hat{\mathcal{U}}} \in \mathbf{X}$ be arbitrary. Then it is not difficult to see that weak form (43) involves only terms of the form $D_i uvw$ and $uvwz$, where u, v, w, and z belong to $H^1(\Omega)$. Thus, each term can be estimated using (44) or (45):

$$|(\mathrm{g}_1, \underline{\mathrm{V}})| \le C_1(\mathcal{U},\mathcal{U}_0,\lambda)(\|\hat{\mathcal{U}}\|_{\mathbf{X}} + \|\hat{\hat{\mathcal{U}}}\|_{\mathbf{X}})\|\underline{\mathrm{V}}\|_1\,;$$

$$|(\mathrm{g}_2, \mathrm{u})| \le C_2(\mathcal{U},\mathcal{U}_0,\lambda)(\|\hat{\mathcal{U}}\|_{\mathbf{X}} + \|\hat{\hat{\mathcal{U}}}\|_{\mathbf{X}})\|\mathrm{u}\|_1\,;$$

$$|(\mathrm{g}_3, q)| \le C_3(\mathcal{U},\mathcal{U}_0,\lambda)(\|\hat{\mathcal{U}}\|_{\mathbf{X}} + \|\hat{\hat{\mathcal{U}}}\|_{\mathbf{X}})\|q\|_1\,;$$

where C_i is polynomial function of $\|\mathcal{U}\|_{\mathbf{X}}$, $\|\mathcal{U}_0\|_{\mathbf{X}}$, and the parameter λ. It then follows that $D^2_{\mathcal{U}}G(\lambda,\mathcal{U})$ is bounded in the norm of $L(\mathbf{X}, L(\mathbf{X},\mathbf{Y}))$ on all bounded subsets of $\Lambda \times \mathbf{X}$. □

This completes verification of all assumptions of Theorem 1. As a result, we can conclude that error estimates (36) and (37) hold for the least-squares finite element approximation as long as problem (22) has a regular branch of solutions with sufficient regularity.

WELL-POSEDNESS OF THE LEAST-SQUARES FORM

In this section, we address the question of the well-posedness of least-squares formulation (22). More precisely, our aim is to show that if $\{(\lambda, (\mathbf{u}(\lambda), p(\lambda))) \mid \lambda \in \Lambda\}$ is a branch of regular solutions of original velocity-pressure Navier-Stokes problem (1)-(4), then

$$\{(\lambda, (\underline{\mathbf{U}}(\lambda), \mathbf{u}(\lambda), p(\lambda))) \mid \lambda \in \Lambda\}$$

is a regular branch for variational problem (22). This is an important question not only because application of Theorem 1 requires a regular branch, but also because it would assert that the least-squares formulation does not introduce bifurcation phenomena that are not already present in the original equations. The question is also nontrivial since the equivalent strong form of (22) now involves derivatives of nonlinear equations (1)-(2).

Assume that $(\mathbf{u}(\lambda), p(\lambda)) \in H_0^1(\Omega)^n \times L_0^2(\Omega)$ yields a regular branch of solutions of (1)-(4), i.e., for every $\lambda \in \Lambda$ the pair $(\mathbf{u}(\lambda), p(\lambda))$ is a nonsingular (weak) solution of the Navier-Stokes equations. We recall the result of [6] that $(\mathbf{u}, p)$ is a nonsingular solution if and only if the linearized problem

$$\begin{aligned} -\nu\triangle\hat{\mathbf{u}} + (\nabla\hat{\mathbf{u}}^t)^t\mathbf{u} + (\nabla\mathbf{u}^t)^t\hat{\mathbf{u}} + \nabla\hat{p} &= \mathbf{f}^* \text{ in } \Omega \\ \nabla^t\hat{\mathbf{u}} &= 0 \text{ in } \Omega \\ \hat{\mathbf{u}} &= \mathbf{0} \text{ on } \Gamma \\ \int_\Omega \hat{p}d\Omega &= 0 \end{aligned}$$

has a unique (weak) solution $(\hat{\mathbf{u}}, \hat{p}) \in H_0^1(\Omega)^n \times L_0^2(\Omega)$ for each $\mathbf{f}^* \in H^{-1}(\Omega)^n$. Specialized to our needs, the nonsingularity assumption asserts that the problem

$$-\nu\triangle\hat{\mathbf{u}} + (\nabla\hat{\mathbf{u}}^t)^t(\mathbf{u} + \mathbf{u}_0) + (\nabla(\mathbf{u}^t + \mathbf{u}_0^t))^t\hat{\mathbf{u}} + \nabla\hat{p} = \mathbf{f}^* \text{ in } \Omega \quad (48)$$

$$\nabla^t\hat{\mathbf{u}} = 0 \text{ in } \Omega \quad (49)$$

$$\hat{\mathbf{u}} = \mathbf{0} \text{ on } \Gamma \quad (50)$$

$$\int_\Omega \hat{p}d\Omega = 0 \quad (51)$$

has a unique (weak) solution $(\hat{\mathbf{u}}, \hat{p}) \in H_0^1(\Omega)^n \times L_0^2(\Omega)$ for each $\mathbf{f}^* \in H^{-1}(\Omega)^n$, where $(\mathbf{u}_0, p_0)$ solves Stokes problem (12) with the original data $\mathbf{f}$.

Under this assumption, well-posedness of (22) will follow if we can establish that $\mathcal{U}(\lambda) = (\underline{\mathbf{U}}(\lambda), \mathbf{u}(\lambda), p(\lambda))$ with $\underline{\mathbf{U}}(\lambda) = \nabla\mathbf{u}(\lambda)^t$ is a nonsingular solution of (22) for all $\lambda \in \Lambda$. In terms of canonical representation (26), this amounts to showing that the linearized mapping $D_{\mathcal{U}}F(\lambda, \mathcal{U})$ is an isomorphism of $\mathbf{X}$; that is, the linearized equation

$$D_{\mathcal{U}}F(\lambda, \mathcal{U})\hat{\mathcal{U}} = (I + T \cdot D_{\mathcal{U}}G(\lambda, \mathcal{U}))\hat{\mathcal{U}} = \mathcal{V} \quad (52)$$

has a unique solution $\hat{\mathcal{U}} \in \mathbf{X}$ for all $\mathcal{V} \in \mathbf{X}$.

Compactness of $T : \mathbf{Z} \mapsto \mathbf{X}$ follows from (35), which asserts that it is a uniform limit of compact operators T_h. Now, from Lemma 5, we have $D_{\mathcal{U}}G(\lambda, \mathcal{U}) \in L(\mathbf{X}, \mathbf{Z})$, so the operator $T \cdot D_{\mathcal{U}}G(\lambda, \mathcal{U}) : \mathbf{X} \mapsto \mathbf{X}$ is compact. Thus, the Fredholm alternative can be applied to (52), and we can assert that $D_{\mathcal{U}}F(\lambda, \mathcal{U})$ is indeed an isomorphism of $\mathbf{X}$ if and only if the homogeneous equation

$$D_U F(\lambda, U)\hat{U} = (I + T \cdot D_U G(\lambda, U))\hat{U} = 0 \tag{53}$$

has only the trivial solution $\hat{U} = 0$ in $\mathbf{X}$. This fact is established in the next lemma using our nonsingularity assumption on $(\mathrm{u}(\lambda), p(\lambda))$.

Lemma 6. *Assume that* (u, p) *is such that linearized equations (48)-(51) have a unique solution for each* $\mathrm{f}^* \in H^{-1}(\Omega)^n$. *Then homogeneous problem (53) has only the trivial solution.*

Proof. Using definitions (31) and (33), one can easily verify that (53) is equivalent to the variational problem

Find $(\underline{\hat{\mathrm{U}}}, \hat{\mathrm{u}}, \hat{p}) \in \mathbf{X}$ such that

$$\begin{aligned}
&\mathcal{B}((\underline{\hat{\mathrm{U}}}, \hat{\mathrm{u}}, \hat{p}), (\underline{\mathrm{V}}, \mathrm{v}, p)) = \\
&\quad \Big(-(\nabla^t \underline{\hat{\mathrm{U}}})^t + \frac{1}{\nu}((\underline{\mathrm{U}} + \underline{\mathrm{U}}_0)^t \hat{\mathrm{u}} + \underline{\hat{\mathrm{U}}}^t(\mathrm{u} + \mathrm{u}_0)) + \nabla \hat{p}, \\
&\qquad -(\nabla^t \underline{\mathrm{V}})^t + \frac{1}{\nu}\left((\underline{\mathrm{U}} + \underline{\mathrm{U}}_0)^t \mathrm{v} + \underline{\mathrm{V}}^t(\mathrm{u} + \mathrm{u}_0)\right) + \nabla q\Big)_0 + \\
&\quad \left(\nabla^t \hat{\mathrm{u}}, \nabla^t \mathrm{v}\right)_0 + \left(\nabla(tr\underline{\hat{\mathrm{U}}}), \nabla(tr\underline{\mathrm{V}})\right)_0 \\
&\quad \left(\underline{\hat{\mathrm{U}}} - \nabla \hat{\mathrm{u}}^t, \underline{\mathrm{V}} - \nabla \mathrm{v}^t\right)_0 + \left(\nabla \times \underline{\hat{\mathrm{U}}}, \nabla \times \underline{\mathrm{V}}\right)_0 = 0
\end{aligned} \tag{54}$$

for all $(\underline{\mathrm{V}}, \mathrm{v}, q) \in \mathbf{X}$.

Variational problem (54) is evidently the Euler-Lagrange equation for the minimization problem

Find $(\underline{\hat{\mathrm{U}}}, \hat{\mathrm{u}}, \hat{p}) \in \mathbf{X}$ such that

$$J_l(\underline{\hat{\mathrm{U}}}, \hat{\mathrm{u}}, \hat{p}) \le J_l(\underline{\mathrm{V}}, \mathrm{v}, q) \quad \text{for all } (\underline{\mathrm{V}}, \mathrm{v}, q) \in \mathbf{X}, \tag{55}$$

where

$$\begin{aligned}
J_l(\underline{\hat{\mathrm{U}}}, \hat{\mathrm{u}}, \hat{p}) &= \| -(\nabla^t \underline{\hat{\mathrm{U}}})^t + \frac{1}{\nu}((\underline{\mathrm{U}} + \underline{\mathrm{U}}_0)^t \hat{\mathrm{u}} + \underline{\hat{\mathrm{U}}}^t(\mathrm{u} + \mathrm{u}_0)) + \nabla \hat{p}\|_0^2 \\
&+ \|\nabla^t \hat{\mathrm{u}}\|_0^2 + \|\underline{\hat{\mathrm{U}}} - \nabla \hat{\mathrm{u}}^t\|_0^2 \\
&+ \|\nabla(tr\underline{\hat{\mathrm{U}}})\|_0^2 + \|\nabla \times \underline{\hat{\mathrm{U}}}\|_0^2
\end{aligned} \tag{56}$$

Thus, nonsingularity of $(\underline{\mathrm{U}}, \mathrm{u}, p)$ would follow if we could show that (55) has no nontrivial minimizers. Assume the contrary. Then the nontrivial minimizer $(\underline{\hat{\mathrm{U}}}, \hat{\mathrm{u}}, \hat{p})$ satisfies

$$-(\nabla^t \underline{\hat{\mathrm{U}}})^t + \frac{1}{\nu}((\underline{\mathrm{U}} + \underline{\mathrm{U}}_0)^t \hat{\mathrm{u}} + \underline{\hat{\mathrm{U}}}^t(\mathrm{u} + \mathrm{u}_0)) + \nabla \hat{p} = 0 \tag{57}$$

$$\underline{\hat{\mathrm{U}}} - \nabla \hat{\mathrm{u}}^t = \underline{0} \tag{58}$$

$$\nabla^t \hat{\mathrm{u}} = 0. \tag{59}$$

Then from equations (57) and (58) and identities $\underline{\mathrm{U}} = \nabla \mathrm{u}^t$ and $\underline{\mathrm{U}}_0 = \nabla \mathrm{u}_0^t$, we conclude that the pair $(\hat{\mathrm{u}}, \hat{p})$ satisfies

$$-\nu\triangle\hat{\mathrm{u}} + (\nabla(\mathrm{u}^t + \mathrm{u}_0^t))^t\hat{\mathrm{u}} + (\nabla\hat{\mathrm{u}}^t)^t(\mathrm{u} + \mathrm{u}_0) + \nabla\hat{p} = \mathbf{0}\,.$$

Now the premise that $(\hat{\underline{\mathrm{U}}}, \hat{\mathrm{u}}, \hat{p})$ is nontrivial, together with (58), implies that $(\hat{\mathrm{u}}, \hat{p})$ is nontrivial. Since (59) is also satisfied, then $(\hat{\mathrm{u}}, \hat{p})$ is also a nontrivial solution of (48)-(51), which is a contradiction. □

MULTIGRID SOLVER FOR THE DISCRETE SYSTEM

Here we consider a simple iterative method applied to (27) and show that it converges linearly with bound uniform in h and λ. Our approach rests on using a multigrid preconditioner for T_h and observing that the operator in (27) is well-conditioned uniformly in h and λ. The development is greatly simplified by basing the analysis on the inner product $\mathcal{B}_S(\cdot,\cdot)$ defined in (31) and by choosing elements of the multigrid-based algorithm that are very easy to analyze. (Most assumptions are made only for convenience; more general conditions can be handled with more cumbersome but straightforward arguments. However, allowing for the more effective *direct* treatment of the nonlinearity within the multigrid process would require much more sophisticated analysis tools than we use here.)

Let M_h be defined so that $\mathcal{U}^h = M_h\mathrm{g}$ represents one or more cycles of (additive or multiplicative) multigrid applied to problem (32), starting from the initial guess $\mathcal{U}^h = 0$. For simplicity, assume that M_h is symmetric in the $\mathcal{B}_S(\cdot,\cdot)$ inner product (e.g., M_h may consist of one relaxation of point Gauss-Seidel with a given ordering before coarsening and one relaxation with the reverse ordering afterwards). Again for simplicity, assume that M_h is so effective that

$$\delta\mathcal{B}_S(T_h\mathcal{V}^h, \mathcal{V}^h) \le \mathcal{B}_S(M_h\mathcal{V}^h, \mathcal{V}^h) \le \mathcal{B}_S(T_h\mathcal{V}^h, \mathcal{V}^h) \tag{60}$$

for all $\mathcal{V}^h \in \mathbf{X}_h$ and for some positive constant δ independent of h and λ. The upper bound can be assured simply by dividing the usual multigrid cycle by 2, and the lower bound follows from the product H^1 equivalence of $\mathcal{B}_S(\cdot,\cdot)$ established in [3]. Assume that

$$\{(\lambda,\mathcal{U}(\lambda)) \,|\, \lambda \in \Lambda\}$$

is a branch of regular solutions of (26), and let $F^h(\lambda,\mathcal{U}^h) = 0$ denote canonical form (27). Then it is easy to see that there exists a neighborhood $\mathcal{O}$ of the origin in $\mathbf{X}$ and positive constants γ and ρ, independent of h and λ, such that

$$\gamma\mathcal{B}_S(\mathcal{V}^h, \mathcal{V}^h) \le \mathcal{B}_S(D_{\mathcal{U}}F^h(\lambda,\mathcal{U})\mathcal{V}^h, \mathcal{V}^h) \le \rho\mathcal{B}_S(\mathcal{V}^h, \mathcal{V}^h) \tag{61}$$

for all $\mathcal{V}^h \in \mathbf{X}_h$, where $(\lambda,\mathcal{U})$ is any element of $\Lambda \times \mathbf{X}_h$ for which $\mathcal{U}(\lambda) - \mathcal{U} \in \mathcal{O}$. The lower bound follows from our regular branch assumption, and the upper bound follows from Lemma 2 and property 1 of Lemma 5.

The iterative method that we consider for solving (27) is given by the expression

$$\mathcal{U}^h \leftarrow \mathcal{U}^h - sM_h\nabla J(\mathcal{U}^h), \tag{62}$$

where $J(\mathcal{U}^h)$ is the functional in (19) and $s = \frac{1}{\rho}$. Suppose for the moment that $M_h = T_h$.

Then the proof of local linear convergence of (62) in the $\mathcal{B}_S(\cdot,\cdot)$ norm with linear factor bounded by $\sqrt{1-\frac{\alpha}{\rho}}$ would follow from: linearizing $\nabla J(\mathcal{U}^h)$ about the solution of (27); the relation $T_h \nabla J(\mathcal{U}^h) = F^h(\lambda, \mathcal{U}^h)$; and the symmetry of $D_{\mathcal{U}} F^h(\lambda, \mathcal{U})$ in the $\mathcal{B}_S(\cdot,\cdot)$ inner product. For (62) with general M_h, we can then use (60) to prove local linear convergence in the $\mathcal{B}_S(\cdot,\cdot)$ norm with factor bounded by $\sqrt{1-\frac{\delta\gamma}{\rho}}$.

This establishes optimality of our simple iterative method based on a multigrid Stokes preconditioner. It is straightforward to extend this result to a full-multigrid-like scheme, where an approximation to the solution of the Navier-Stokes equations is achieved with accuracy up to discretization error at the cost of a few fine grid operator evaluations.

REFERENCES

[1] P. Bochev; Analysis of least-squares finite element methods for the Navier-Stokes equations, submitted.

[2] P. Bochev and M.D. Gunzburger; Analysis of least-squares finite element methods for the Stokes equations. *Math. Comp.*, 63/108, 1994, pp. 479-506.

[3] Z.Cai, T.A. Manteuffel, and S. F. McCormick; First-order system least-squares for the Stokes equations, submitted.

[4] Z.Cai, T.A. Manteuffel, and S. F. McCormick; First-order system least-squares for partial differential equations: Part II, *SIAM J. Numer. Anal.*, to appear.

[5] C.L. Chang; A mixed finite element method for the Stokes problem: an acceleration pressure formulation, *Appl. Math. Comp.*, 36, 1990, pp.135-146.

[6] V. Girault and P.-A. Raviart; *Finite Element Methods for Navier-Stokes Equations*, Springer, Berlin, 1986.

[7] R. Témam; *Nonlinear Functional Analysis and Navier-Stokes Equations*, SIAM, Philadelphia, 1983.

Page intentionally left blank

MGLab: AN INTERACTIVE MULTIGRID ENVIRONMENT

James Bordner
Faisal Saied
Department of Computer Science
University of Illinois at Urbana-Champaign

SUMMARY

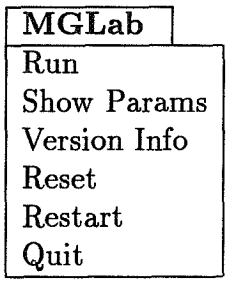
MGLab is a set of Matlab functions that defines an interactive environment for experimenting with multigrid algorithms. The package solves two-dimensional elliptic partial differential equations discretized using either finite differences or finite volumes, depending on the problem. Built-in problems include the Poisson equation, the Helmholtz equation, a convection-diffusion problem, and a discontinuous coefficient problem. A number of parameters controlling the multigrid V-cycle can be set using a point-and-click mechanism. The menu-based user interface also allows a choice of several Krylov subspace methods, including CG, GMRES(k), and Bi-CGSTAB, which can be used either as stand-alone solvers or as multigrid acceleration schemes. The package exploits Matlab's visualization and sparse matrix features and has been structured to be easily extensible.

WHAT IS MGLab?

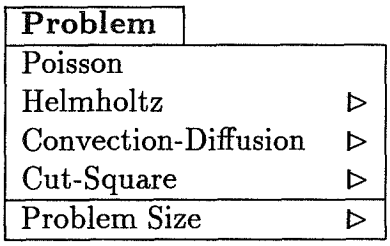
MGLab is an interactive environment based on Matlab Version 4.0 for solving elliptic partial differential equations using multigrid algorithms. A graphical user interface (GUI) enables the user to select a problem, set parameters for the multigrid V-cycle, optionally choose a Krylov subspace accelerator, and visualize the results. MGLab is written in Matlab [1], which has greatly simplified the programming but has led to some loss of efficiency in a few respects.

A number of very good introductions to multigrid methods are available that can be used in conjunction with MGLab, including refs. [2]–[4]. The numerical treatment of elliptic partial differential equations is discussed in ref. [5], and the finite volume method for discretizing elliptic problems is described in ref. [6]. Some of the experiments described in ref. [7] have been included in MGLab as demos. Some software that addresses similar issues is described in refs. [8]–[14]. The basic linear algebra concepts needed for a number of the components of MGLab, including the iterative solvers, are discussed in refs. [15]–[23]. Other references that may be useful background reading for MGLab users include [24]–[34].

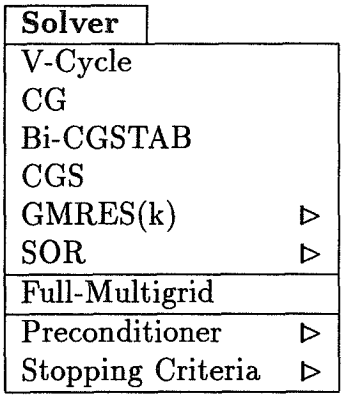
The interface between MGLab and the user is a menu structure; the menu items can be selected using a point-and-click mechanism. Menu items are grouped according to their function, depending on whether they relate to the partial differential equation, the solver, multigrid parameters, visualization of results, or built-in demos. Top level menu choices and their submenus are outlined below.



The submenus in the MGLab top-level menu item control the basic behavior of the package, including solving the currently selected problem, displaying the currently selected parameters, and restarting MGLab with the default parameters.



The submenus in the Problem top-level menu item select which partial differential equation to solve; further submenus are available for setting problem-dependent parameters. The problem size can also be set here.



The submenus in the Solver top-level menu item are used to select the solver, choose a preconditioner if desired, and set the stopping criteria. The GMRES menu item has a submenu for choosing the GMRES restart parameter, and the SOR menu item has a submenu for choosing the SOR relaxation parameter.

MG-Parameters	
Number of Levels	▷
Smoother	▷
Restriction	▷
Prolongation	▷
Coarse-grid Solver	▷
Coarse-grid Operator	▷
MG Cycle	▷

The submenus in the MG-Parameters top-level menu item are used to set various multigrid parameters, including the number of grid levels, the smoother, the restriction and prolongation operators, the solver for the coarse grid problem, the method for generating the operators on the coarser grids, and the type of multigrid cycle, such as the V-cycle or the W-cycle.

Visualize	
Convergence History	
Computed Solution (surf)	
Computed Solution (pcolor)	
X-Axis	▷
Y-Axis	▷

The submenus in the Visualize top-level menu item are used to view the results after solving a problem. The convergence history can be plotted, the scaling along the x and y axes for the convergence history plot can be chosen, and the numerical solution can be displayed either as a surface plot or a contour plot.

Demos
Smoothers
Fourier analysis
Truncation error

The submenus in the Demos top-level menu item select and run demonstrations that illustrate specific properties of multigrid methods, such as the behavior of different smoothers, how the errors after the coarse grid correction and after the post-smoothings in the V-cycle behave in physical and Fourier space, and how the truncation error compares with the discrete residual.

ELLIPTIC PROBLEMS

The built-in test problems in the current version of MGLab are restricted to two-dimensional elliptic partial differential equations on rectangular domains.

$$\begin{aligned} \nabla \cdot (a\nabla u) + b \cdot \nabla u + cu = f \quad \text{in} \quad \Omega, \\ u = g \quad \text{on} \quad \partial\Omega. \end{aligned} \tag{1}$$

The domain Ω is the unit square $\{(x,y) : 0 < x,y < 1\}$, and the elliptic problem is discretized using the standard 5-point stencil on a uniform mesh. Currently the test problems all have zero Dirichlet boundary conditions. The matrices are stored using Matlab's sparse storage format, which ensures that matrix-vector products are efficient. Furthermore, the coarse grid problem can be solved using Matlab's built-in sparse direct solver [35], which uses graph-theoretic techniques to reorder the rows and columns of the matrix to reduce fill-in during the elimination process.

Even within the discretization and boundary condition restrictions, a number of different types of elliptic problems are possible. MGLab's test suite includes the Poisson equation, the Helmholtz equation, a convection-diffusion equation, and a discontinuous coefficient problem ("cut-square").

Poisson Equation. The Poisson equation, $-\nabla^2 u = f$, is the easiest problem in MGLab to solve; the coefficient matrix of the discretized equation is both symmetric and positive definite.

Helmholtz Equation. The Helmholtz equation, $-\nabla^2 u + ku = f$, is the same as the Poisson equation, except for the ku term. Depending on k, this term can make the problem indefinite or complex. The parameter k can be selected by the user, where $k \in \{-10, -5, -1, 0, 1, 5, 10, 10 + i\}$.

Convection-Diffusion Equation. The convection-diffusion equation, $-\nabla^2 u + \lambda u_x + \sigma u = f$, adds the convection term λu_x to the Helmholtz equation. This added term can make the coefficient matrix of the discretized problem nonsymmetric. The parameters λ and σ can be selected by the user, where $\lambda \in \{0, 10, 100, 1000\}$ and $\sigma \in \{-100, -50, 0, 5, 10,$
$20, 50, 100\}$.

Cut-Square Equation. The cut-square equation is $-\nabla \cdot (a \nabla u) = f$, where a is a discontinuous function of x and y. Specifically, $a(x, y) = \alpha$ for $0.4 \leq x, y \leq 0.6$, and $a(x, y) = 1$ elsewhere in Ω. The parameter α can be selected by the user, where $\alpha \in \{0.001, 0.01, 0.1, 1, 10, 100, 1000\}$.

MULTIGRID PARAMETERS

MGLab is designed to solve elliptic partial differential equations using multigrid methods, with the option to embed the multigrid solver as a preconditioner in a Krylov subspace method. A number of parameters that determine the V-cycle can be set through the graphical user interface. These include the number of levels, the smoother, the number of pre- and post-smoothing sweeps, the restriction and prolongation operators, the coarse grid solver, and the type of multigrid cycle.

Number of Levels. The number of grid levels can be chosen to be between 1 and 5. Note that `levels` $= 1$ corresponds to a sparse direct solver. If the chosen number of levels is too large for the current problem size, it is set to the largest number possible.

Smoothers. The available smoothers are weighted Jacobi, Gauss-Seidel, and Red/Black Gauss-Seidel. For the Jacobi smoother, the user can pick the weighting factor. The number of pre- and post-smoothing sweeps (ν_1, ν_2) can also be set through the Smoother submenu.

Restriction Operators. The restriction operators available in MGLab are injection, half-weighting, and full-weighting. These are implemented with fairly compact code that uses Matlab's colon notation for accessing arrays.

Prolongation Operators. MGLab offers bilinear and cubic interpolation as the choices for prolongation. These are currently implemented by calls to Matlab's `interp2` function.

Coarse Grid Solver. The default coarse grid solver is Matlab's built-in sparse direct solver [35]. As an alternative, the user can choose to use the smoother as the coarse grid solver. This is less costly but also less accurate.

Multigrid Cycle. Although the V-cycle is the default multigrid cycle, the user can also select the W-cycle. Other cycles, such as the half V-cycle or weighted V-cycle, could be added easily. Full multigrid can be selected in the Solver menu.

KRYLOV SUBSPACE ACCELERATORS

The V-cycle defined through the multigrid parameters discussed previously can be used as an iterative solver on its own or as a preconditioner for Krylov subspace methods, such as CG, GMRES(k), and Bi-CGSTAB. For solving the linear system of equations $Ax = b$, these methods work with a sequence of Krylov subspaces defined by

$$K_j(r_0, A) = \text{span}\{r_0, Ar_0, \ldots, A^{j-1}r_0\}. \tag{2}$$

The j-th iterate x_j is picked from

$$x_j \in x_0 + K_j(r_0, A),$$

where r_0 is the initial residual $b - Ax_0$.

Below we list some of the important properties of the methods; details of the algorithms can be found in the references given with each method. See also refs. [19] and [22].

CG. The Conjugate Gradient method is a Krylov subspace accelerator for symmetric positive definite (SPD) systems; this method minimizes the A-norm of the error at each iteration. The preconditioned version (PCG) requires a symmetric positive definite preconditioner. The CG method was developed by Hestenes and Stiefel [16] and is discussed in ref. [15].

GMRES(k). The Generalized Minimum Residual method of Saad and Schultz [18] is a direct generalization of the CG method to matrices that are not SPD. The CG method takes advantage of a three-term recurrence relation that is not available in GMRES, so both the number of vectors that must be stored and the number of floating point operations performed increase with each iteration. For this reason, GMRES is typically restarted every k iterations.

CGS. Conjugate Gradient Squared is a variant of the Bi-Conjugate Gradient (Bi-CG) method that, unlike Bi-CG, avoids multiplication by the transpose of the matrix

A. The CGS method was proposed by Sonneveld [21]. Unlike GMRES, this method is not guaranteed to minimize the 2-norm of the residual, but the number of vectors required does not increase with each iteration so CGS does not need to be restarted. The convergence behavior of CGS can be very erratic.

Bi-CGSTAB. This method was introduced by Van der Vorst [20] and is transpose-free like CGS, but with a more regular convergence behavior.

SOR. The Successive Over-Relaxation method [17] is a stationary iterative method with a relaxation parameter ω. If $\omega = 1$, then SOR reduces to the Gauss-Seidel method.

PRECONDITIONERS

The performance of iterative methods can often be enhanced with preconditioning by premultiplying the linear system $Ax = b$ by an approximate inverse M^{-1} of A:

$$M^{-1}Ax = M^{-1}b. \tag{3}$$

The multigrid V-cycle can be used as a preconditioner. In addition, even though our emphasis is on multigrid methods, MGLab allows the V-cycle preconditioner to be replaced by something else. The current preconditioners available in MGLab for the Krylov subspace methods are the V-cycle, point Jacobi, and point Gauss-Seidel methods. Other preconditioners, such as the block Jacobi, red-black Gauss-Seidel, ILU, and SSOR methods, could be added relatively easily.

A standardized interface to the preconditioner is available that is independent of the iterative solver. The operation $z \leftarrow M^{-1}r$ is performed by the following call:

```
z = precondition(A, r).
```

The function `precondition` accesses the parameters needed to apply the preconditioner M^{-1}, which is implicitly defined in terms of A. This enhances the extensibility of MGLab in the sense that adding Matlab implementations of other iterative methods would be straightforward.

VISUALIZATION OPTIONS

MGLab exploits Matlab's powerful graphics capabilities to plot the convergence history of the solution process and to visualize the computed solution. Currently we make use of the `plot`, `surf`, `pcolor`, and `contour` commands in Matlab. The visualization options are available through the graphical user interface.

```
function u_out = vCycle(level, b, u_in)

% Use the zero vector for u_in as the default

if nargin == 2,
   u_in = zeros(size(b));
end

if level == coarsest(level)
   u_out = coarse_grid_solve(level, b);
else
   u       = smooth(level, b, u_in, 'pre');
   r       = residual(level, b, u);
   b_c     = restrict(level, r);
   u_c     = vCycle(level+1, b_c);
   correct = interpolate(level, u_c);
   u       = u + correct;
   u_out   = smooth(level, b, u, 'post');
end
```

Figure 1: V-cycle Function

SOME COMMENTS ON THE INTERNAL STRUCTURE OF MGLab

MGLab is written entirely in Matlab. One group of functions is devoted to the user interface; these make use of Matlab's `uimenu` function. Other groups of functions implement the problem generation, algorithms, and visualization in MGLab.

MGLab makes use of Matlab's `global` mechanism. This approach leads to a considerable simplification of the programming in many situations but carries the software engineering risk of non-transparent code and the danger of subtle bugs. We have attempted to write the higher level functions such as `sp_laplace`, `Vcycle`, `pcg`, and `precondition` in a way that does not require them to see the global variables. This results in very compact and readable code and reduces the chance that global variables will be accidentally damaged. The code for `Vcycle` is shown in Figure 1 to illustrate this approach.

The "middle level" functions, such as `smooth` and `restrict`, access the global workspace in a disciplined manner. Some low level functions were created expressly for the purpose of accessing the globals and returning a single value so that the globals could be hidden from the higher level functions.

BUILT-IN DEMOS IN MGLab

MGLab has a working and extensible framework for adding numerical experiments. Currently, the numerical experiments supplied with MGLab are the following:

- A numerical study of the smoothing properties of the weighted Jacobi, Gauss-Seidel, and Red/Black Gauss-Seidel approaches, in physical and Fourier space [7]. The Fourier transforms are constructed out of Matlab's fast Fourier transform (fft).
- A numerical Fourier analysis of the complementary roles of the coarse grid correction and the smoother for a model problem.
- A comparison of the truncation error (pde error) and the discrete residual [7]. This demo highlights the ability of multigrid methods to achieve truncation error accuracy very rapidly.

Figures 2 through 5 show the output of Demo 2. The intention of this demo is to give a visual sense for the different roles of the coarse grid correction and the (post-)smoothing. In this demo, we solve the Poisson problem on a 49×49 mesh by multigrid. The V-cycle parameters are as follows:

- Two levels
- Gauss-Seidel smoothing
- $(\nu_1, \nu_2) = (0, 4)$, i.e., no pre-smoothing and 4 post-smoothing sweeps
- Half-weighting restriction
- Cubic interpolation
- The coarse grid solver is Matlab's built-in sparse Gaussian elimination

The initial guess was chosen so that the initial error had a mix of low and high frequencies:

$$e^{(0)}(x, y) = \sum_{j=1}^{4} \sin(10j\pi x) \sin(10j\pi y).$$

Figure 2 shows the initial error on the left and the absolute values of the (scaled[1]) Fourier coefficients of the error on the right. Figure 3 shows the error in the first V-cycle, after the coarse grid correction (top) and after the post-smoothing (bottom). In each case, the error is shown in "physical" space (left) and in Fourier space (right).

[1]The 2D sine transform was applied to the error on the mesh.

Figures 4 and 5 are the same as Figures 3 except that in Figures 4 and 5 the errors are shown in the second and third iterations, respectively.

These figures show how the coarse grid correction and the smoother complement each other by reducing the low frequency and high frequency error components, respectively.

OBTAINING AND INSTALLING MGLab

MGLab V1.0 is currently available via anonymous ftp to casper.cs.yale.edu in the directory /mgnet/Codes/mglab. After the tar file is uncompressed, it should be untarred in a subdirectory such as `~myname/matlab/MGLab`. To run MGLab, simply change to your MGLab directory, start up Matlab and type `MGLab`.

Comments and suggestions for improvements to the code are welcome; we plan to release future versions of MGLab that incorporate enhancements and bug fixes.

ACKNOWLEDGMENTS

The work of the second author was supported in part by an NSF Research Initiation Award, NSF ASC 92 09502 RIA.

REFERENCES

[1] The MathWorks, Inc., *Matlab Reference Guide*, Natick, Mass., 1992.

[2] D. C. Jespersen. Multigrid methods for partial differential equations. In G. H. Golub, editor, *Studies in Numerical Analysis*, The Mathematical Association of America, pp. 270–318, 1984.

[3] W. L. Briggs. *A Multigrid Tutorial.* SIAM, Philadelphia, 1988.

[4] A. Brandt. Multilevel adaptive solutions to boundary-value problems. *Math. Comp.*, 31:311–329, 1977.

[5] G. Birkhoff and R. E. Lynch. *Numerical Solution of Elliptic Problems.* SIAM, Philadelphia, 1984.

[6] R. S. Varga. *Matrix Iterative Analysis.* Prentice-Hall, New Jersey, 1962.

[7] F. Saied and M. J. Holst. Vector multigrid: An accuracy and performance study. Technical Report UIUCDCS-R-90-1636, Department of Computer Science, University of Illinois at Urbana-Champaign, 1990.

[8] R. E. Bank. *PLTMG: A Software Package for Solving Elliptic Partial Differential Equations. Users' Guide 7.0.* SIAM, Philadelphia, 1994.

[9] W. F. Mitchell. MGGHAT: Elliptic PDE software with adaptive refinement, multigrid and high order finite elements. In N. D. Melson, T. A. Manteuffel, and S. F. McCormick, editors, *Proceedings of the Sixth Copper Mountain Conference on Multigrid Methods, Part 2*, pp. 439–448, NASA, 1993.

[10] W. Gropp and B. Smith. Simplified linear equation solvers users manual. Technical Report ANL-93/8, MCS Division, Argonne National Laboratory, 1993.

[11] W. Gropp and B. Smith. User's manual for KSP data-structure-neutral codes implementing Krylov space methods. Technical Report ANL-93/30, MCS Division, Argonne National Laboratory, 1993.

[12] J. R. Rice and R. F. Boisvert. *Solving Elliptic Problems using ELLPACK.* Springer-Verlag, New York, 1985.

[13] B. Smith. Extensible PDE solvers package user's manual. Technical Report ANL-94/40, MCS Division, Argonne National Laboratory, 1994.

[14] Y. Saad. SPARSKIT: A basic toolkit for sparse matrix computations. Technical Report 1029, CSRD, University of Illinois at Urbana-Champaign, 1990.

[15] G. H. Golub and C. F. Van Loan. *Matrix Computations.* Johns Hopkins University Press, Baltimore, 2nd edition, 1989.

[16] M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *J. Res. Nat. Bur. Stand.*, 49:409–435, 1952.

[17] D. M. Young. *Iterative Solution of Large Linear Systems.* Academic Press, New York, 1971.

[18] Y. Saad and M. H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 7(3):856–869, 1986.

[19] R. W. Freund, G. H. Golub, and N. Nachtigal. Iterative solution of linear systems. *Acta Numerica*, 1:57–100, 1991.

[20] H. A. van der Vorst. Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of non-symmetric problems. *SIAM J. Sci. Stat. Comput.*, 13:631–645, 1992.

[21] P. Sonneveld. CGS: A fast Lanczos-type solver for nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 10:36–52, 1989.

[22] R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, H. Van der Vorst, *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*, SIAM Publications, Philadelphia, 1993.

[23] W. Hackbusch. *Iterative Lösung grosser schwachbesetzter Gleichungssysteme.* Teubner, Stuttgart, 1993.

[24] W. Hackbusch. *Elliptic Differential Equations.* Springer-Verlag, New York, 1992.

[25] A. Brandt. Multigrid solvers on parallel computers. In M. H. Schultz, editor, *Elliptic Problem Solvers.* Academic Press, New York, 1981.

[26] J. E. Dendy, Jr. Black box multigrid. *J. Comp. Phys.*, 48:366–386, 1982.

[27] H. Foerster, K. Stueben, and U. Trottenberg. Non-standard multigrid techniques using checkered relaxation and intermediate grids. In M. H. Schultz, editor, *Elliptic Problem Solvers*, Academic Press, New York, 1981.

[28] W. Hackbusch and U. Trottenberg. *Multi-grid Methods.* Springer-Verlag, Berlin, 1982.

[29] W. Hackbusch. *Multi-grid Methods and Applications.* Springer-Verlag, Berlin, 1985.

[30] M. J. Holst and F. Saied. Parallel performance of some multigrid solvers for three-dimensional parabolic equations. Technical Report UIUCDCS-R-91-1697, Dept. of Computer Science, University of Illinois at Urbana-Champaign, 1991.

[31] M. Holst and F. Saied. Multigrid solution of the Poisson-Boltzmann equation. *J. Comput. Chem.*, 14(1):105–113, 1993.

[32] M. Holst, R. Kozack, F. Saied, and S. Subramaniam. Treatment of electrostatic effects in proteins: Multigrid-based Newton iterative method for solution of the full nonlinear Poisson-Boltzmann equation. *Proteins: Structure, Function, and Genetics*, 18(3):231–245, 1994.

[33] S. McCormick, editor. *Multigrid Methods.* SIAM, Philadelphia, 1987.

[34] F. Saied and M. J. Holst. Multigrid methods for computational acoustics on vector and parallel computers. In R. L. Lau and D. Lee, editors, *Proceedings of the Third IMACS Symposium on Computational Acoustics*, Harvard University, pp. 71–80, 1991.

[35] J. R. Gilbert, C. Moler, and R. Schreiber. Sparse matrices in MATLAB: Design and Implementation. *SIAM J. Mat. Anal. Appl.*, 13:333–356, 1992.

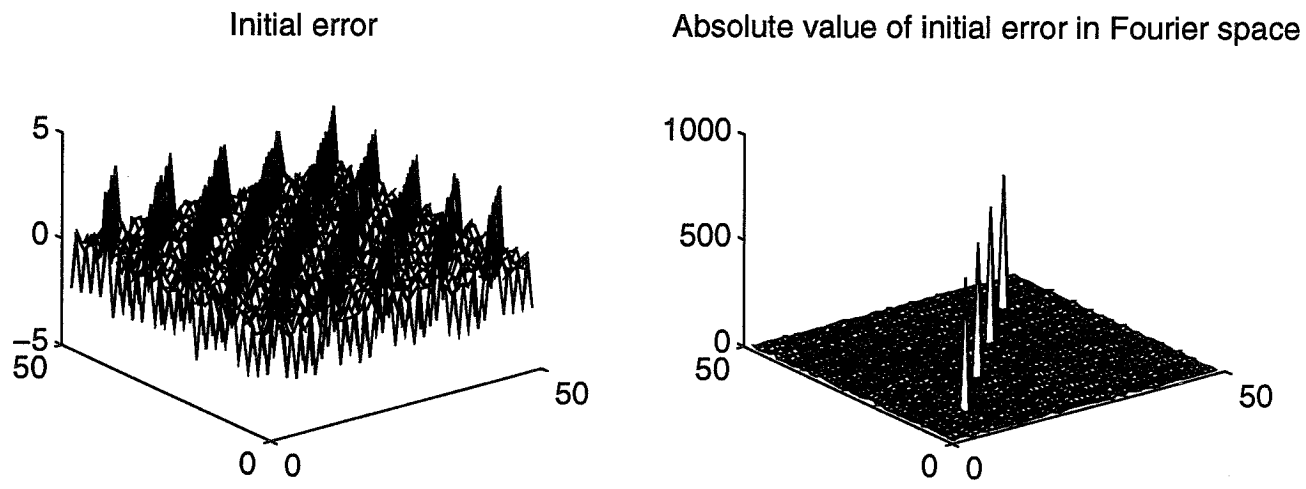


Figure 2: Output from Demo 2. Initial error for Poisson's equation on a 49×49 grid. The two-grid algorithm was used, with Gauss-Seidel smoothing with $(\nu_1, \nu_2) = (0, 4)$, half-weighting, and cubic interpolation. The error is shown on the left, and the 2D sine transform of the error on the right.

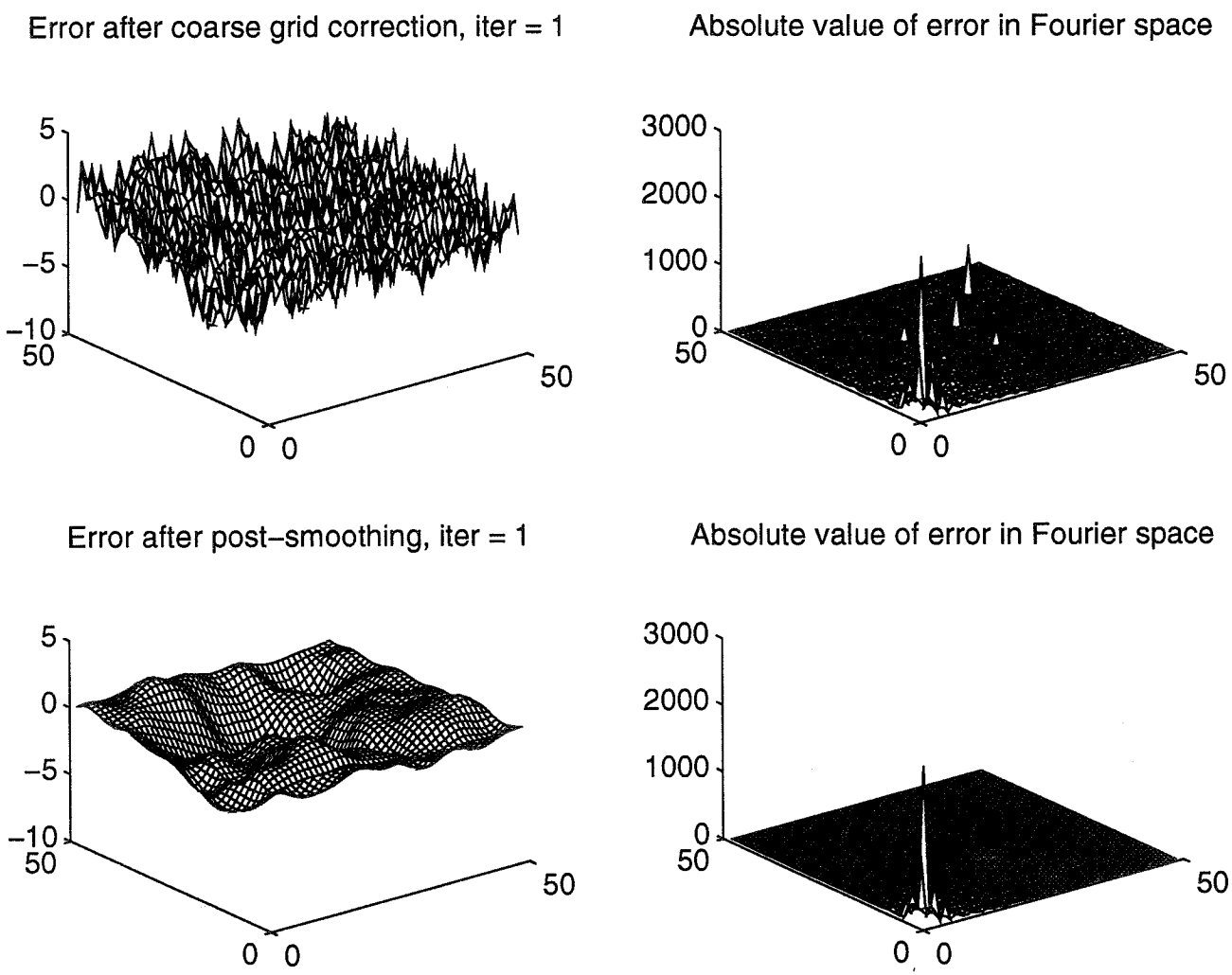


Figure 3: Output from Demo 2. Error in the first V-cycle, after the coarse grid correction (top) and after the post-smoothing (bottom). As in Figure 2, the error in physical space is shown on the left, and the error in Fourier space is shown on the right.

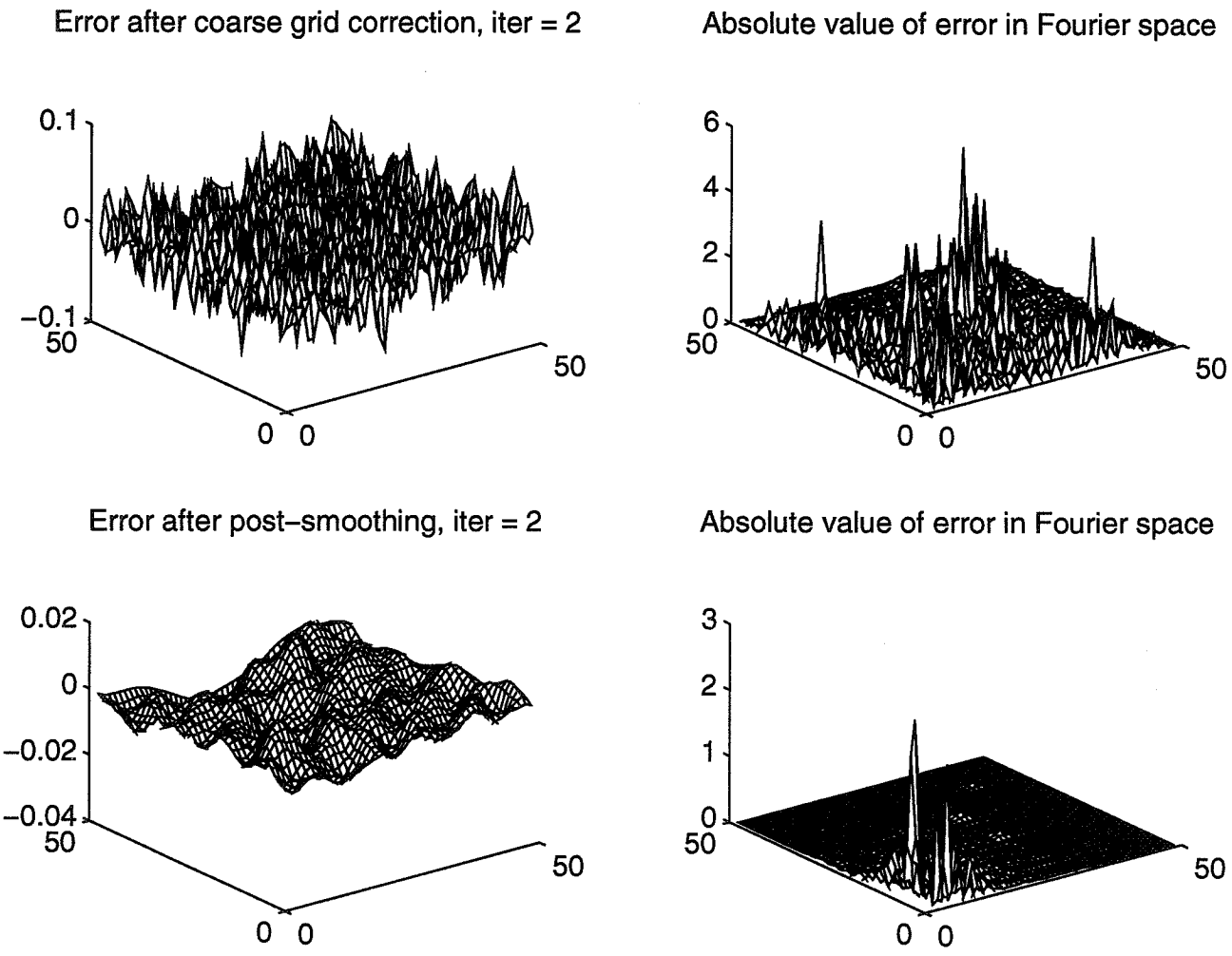


Figure 4: Output from Demo 2 (same as Figure 3, for the second V-cycle).

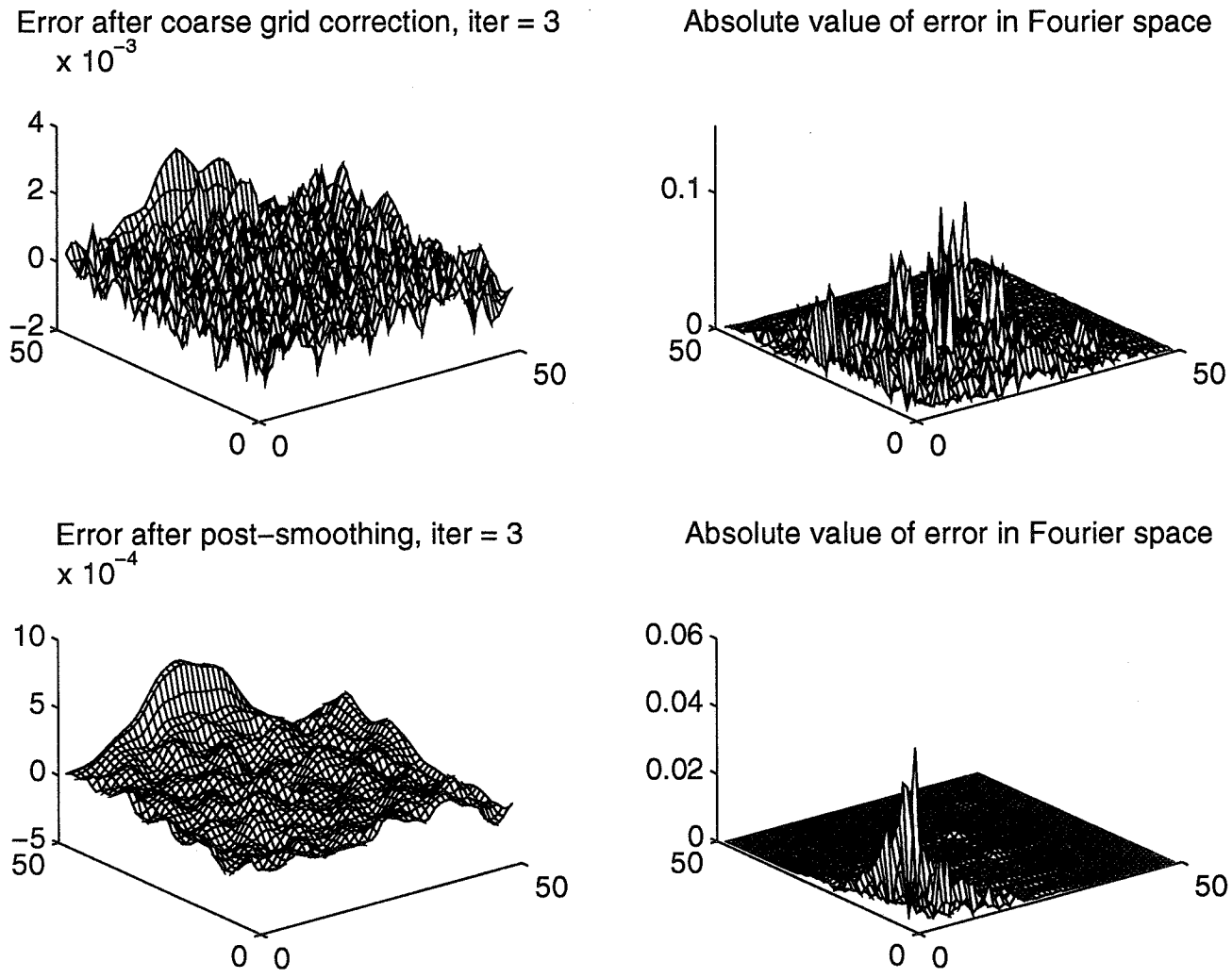


Figure 5: Output from Demo 2 (same as Figure 3, for the third V-cycle).

Page intentionally left blank

A FULL MULTI-GRID METHOD FOR THE SOLUTION OF THE CELL VERTEX FINITE VOLUME CAUCHY–RIEMANN EQUATIONS*

A. Borzì, K.W. Morton, E. Süli, and M. Vanmaele
Oxford University Computing Laboratory
Numerical Analysis Group
Wolfson Building, Parks Road
Oxford, England OX1 3QD

SUMMARY

The system of inhomogeneous Cauchy–Riemann equations defined on a square domain and subject to Dirichlet boundary conditions is considered. This problem is discretised by using the cell vertex finite volume method on quadrilateral meshes. The resulting algebraic problem is overdetermined and the solution is defined in a least squares sense. By this approach a consistent algebraic problem is obtained which differs from the original one by $O(h^2)$ perturbations of the right-hand side.

A suitable cell-based convergent smoothing iteration is presented which is naturally linked to the least squares formulation. Hence, a standard multi-grid algorithm is reported which combines the given smoother and cell-based transfer operators. Some remarkable reduction properties of these operators are shown.

A full multi-grid method is discussed which solves the discrete problem to the level of truncation error by employing one multi-grid cycle at each current level of discretisation.

Experiments and applications of the full multi-grid scheme are presented.

INTRODUCTION

We discuss a full multi-grid algorithm for the numerical solution of the system of inhomogeneous Cauchy–Riemann equations. This algorithm has been formulated in [1]. The Cauchy–Riemann equations are discretised by using a cell vertex finite volume method. We consider the continuous problem defined on a square subject to Dirichlet boundary conditions. Square cells are used for the discretisation.

The motivation for the study of the Cauchy-Riemann system is that it provides a suitable model problem to develop a general multi-grid method for the solution of elliptic flow equations when they are discretised by using a cell vertex finite volume scheme. In this respect, the Cauchy–Riemann equations are the first model in the hierarchy of these fluid flow problems. In particular, it has been clearly shown that the elliptic part of the inviscid incompressible Euler problem is given by the set of Cauchy–Riemann equations [2]. Thus, for example, the present algorithm combined with an appropriate hyperbolic solver would provide an efficient solution method for that inviscid flow.

*This work was financed in part by HCM contract CHRX-CT93-0042 and in part by SERC.

In fact, this idea has been pursued since the work of Brandt and Dinar [3] (see, for example, [4, 5, 2, 6]), where the Cauchy–Riemann equations are taken as a first example of an elliptic system. In [3], for this model problem, a full multi-grid method is developed. Then, the techniques developed for this case are extended to the steady-state Stokes equations and the incompressible Navier-Stokes equations.

However, such methods are constructed to approximate elliptic equations discretised on staggered grids. On the other hand, we want efficient algorithms which solve fluid flow problems discretised by using cell vertex finite volume schemes. Hence the need to re-develop the multi-grid method for problems resulting from the cell vertex discretisation. This is not a mere adaptation of the known techniques, since the peculiarity of the cell vertex scheme renders the previous methods unsuitable for the present task.

In fact, when a cell vertex finite volume discretisation is used, there is generally the problem of how to define a suitable iterative scheme. In a cell vertex approach the resulting equations are cell-based, while the unknowns are node-based. Therefore there is not a one-to-one correspondence between unknowns and equations which can be inverted to provide a node-based iterative scheme. To circumvent this problem the so-called *Kaczmarz* iterative scheme was proposed [7, 8], which was applied in [9], but it proved inefficient as a solver. The use of the Kaczmarz relaxation is natural in the context of cell vertex discretisation. In fact, this type of approximation, when applied to a first order elliptic system, usually results in an overdetermined system for which a least squares approach is necessary. The Kaczmarz iteration is then equivalent to the Gauss–Seidel method applied to the normal equations. We know, in addition, that this relaxation method can be used as a smoother in a multi-grid algorithm (see, for example, [10, 11]).

It is also interesting to compare the Kaczmarz relaxation with another, widely used, method to iteratively solve a cell vertex system of equations, that is, the generalised Lax–Wendroff scheme. See, e.g., [12]. This technique is based on time-stepping the (artificial) unsteady problem, derived from the original one by adding a partial derivative in time of the unknown variables. Then, on each node, a new value for the solution vector is obtained from the previous one by Taylor series expansion in time, up to second order terms. We claim that the second order term in this Taylor expansion is equivalent to the Kaczmarz relaxation.

For these reasons we shall employ a Kaczmarz relaxation as a smoother and combine it with a cell-based transfer operator of the residuals and a node-based prolongation operator of the unknown variables to obtain a fast multi-grid solver.

In the next section we define the differential problem to be solved, that is, the inhomogeneous Cauchy–Riemann equations on a square, subject to Dirichlet boundary conditions. We discretise this problem by using the cell vertex finite volume method on a square mesh. The resulting linear system is handled through a least squares approach. This method is then used in the third section in order to develop an iterative scheme which is known as the Kaczmarz relaxation. In the fourth section, this iterative method is used in combination with a cell-based residual transfer operator, in a multi-grid (MG) cycle. As shown in the section of numerical experiments, the corresponding full multi-grid method solves the discrete problem to the level of the truncation error in just a few work units. Then by using a suitable modification of the Kaczmarz relaxation we shall give an application of the FMG method on non-uniform grids.

THE CAUCHY–RIEMANN EQUATIONS AND THEIR CELL VERTEX DISCRETISATION

We consider the system of Cauchy–Riemann equations

$$\begin{cases} \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = f^{(1)}(x,y) \,, \\ \frac{\partial u}{\partial y} - \frac{\partial v}{\partial x} = f^{(2)}(x,y) \,, \end{cases} \tag{1}$$

in a square domain Ω, where $u(x,y)$ and $v(x,y)$ are the unknown functions, and $f^{(1)} \in L_2(\Omega)$ and $f^{(2)} \in L_2(\Omega)$ represent the source terms. The following Dirichlet boundary conditions are prescribed on the boundary $\partial\Omega$:

$$(u(P), v(P))_n = G(P) \,, \; P = (x,y) \in \partial\Omega \,, \tag{2}$$

where $(u,v)_n$ denotes the component of the vector (u,v) normal to the boundary in the outward direction. The equations (1) with (2) represent a regular elliptic system [13]. The well-posedness of the problem follows from the compatibility condition

$$\int_\Omega f^{(1)} dxdy = \int_{\partial\Omega} G ds \,. \tag{3}$$

If (3) holds then the equations (1), with the boundary conditions (2), have a unique solution.

In order to discretise the problem (1), (2), we assume that the domain Ω is partitioned by a uniform mesh of quadrilateral cells, whose mesh size is h. Each vertex of this grid will be labelled by $_{i,j}$, $i,j = 1,\ldots,N$. We denote $u(x_i,y_j) = u_{i,j}$ and $v(x_i,y_j) = v_{i,j}$, where $x_i = (i-1)*h$ and $y_j = (j-1)*h$. The cell vertex discretisation of the system (1) on these grids follows by integrating the Cauchy–Riemann equations over each cell $\Omega_h^{ij} = [i,i+1] \times [j,j+1]$ and by using Gauss' theorem to convert the integrals into line integrals along the cell edges, which are then discretised using the trapezoidal rule. In this way, the following cell vertex Cauchy–Riemann equations are obtained:

$$\begin{aligned} &\tfrac{1}{2h}(-u_{i,j} - u_{i,j+1} + u_{i+1,j} + u_{i+1,j+1}) + \\ &\tfrac{1}{2h}(-v_{i,j} + v_{i,j+1} - v_{i+1,j} + v_{i+1,j+1}) = \quad \hat{f}^{(1)}_{i,j} \,, \end{aligned} \tag{4}$$

$$\begin{aligned} &\tfrac{1}{2h}(-u_{i,j} + u_{i,j+1} - u_{i+1,j} + u_{i+1,j+1}) + \\ &\tfrac{1}{2h}(v_{i,j} + v_{i,j+1} - v_{i+1,j} - v_{i+1,j+1}) = \quad \hat{f}^{(2)}_{i,j} \,, \end{aligned} \tag{5}$$

where $i,j = 1,\ldots,N-1$ and $\hat{f}^{(l)}_{i,j} = \frac{1}{h^2}\int_{\Omega_h^{ij}} f^{(l)} dxdy$, $l = 1,2$.

The boundary conditions are

$$u_{1,j} = G_{1,j} \,, \; u_{N,j} = G_{N,j} \,, \quad j = 1,\ldots,N \,, \tag{6}$$

$$v_{i,1} = G_{i,1} \,, \; v_{i,N} = G_{i,N} \,, \quad i = 1,\ldots,N \,. \tag{7}$$

For the above cell vertex discretisation we have $2 \times (N \times N)$ unknowns, $2 \times ((N-1) \times (N-1))$ cell equations, $4 \times N$ given boundary values. Therefore we have 2 more equations than unknowns.

By using the compatibility condition it is possible to reduce the number of equations by one. But we still have an overdetermined system.

In the following we will discuss the least squares approach which allows us to define a unique solution to the system of the cell vertex Cauchy-Riemann equations. For this purpose it is convenient to introduce a compact notation. By A we denote the $(2N^2 - 4N + 2) \times (2N^2 - 4N)$ matrix of coefficients, which is derived from (4) and (5). In fact, the first $(N-1)^2$ rows relate to the discrete divergence equation, and the remaining $(N-1)^2$ relate to the curl equation. The boundary values are incorporated in the right-hand side of the system. Thus any element of the right-hand side is of the form $\tilde{f}_{i,j}^{(k)} = \hat{f}_{i,j}^{(k)}$ + boundary values. The right-hand side itself will be denoted by $\tilde{f} = (\tilde{f}^{(1)} \quad \tilde{f}^{(2)})^T$; $\tilde{f}$ is the column vector whose first $(N-1)^2$ elements are the values of $\tilde{f}^{(1)}$ ordered lexicographically, and the last $(N-1)^2$ elements are the values of $\tilde{f}^{(2)}$ ordered in the same way. With this notation the compatibility condition (3) becomes

$$\sum_{i,j=1}^{N-1} \tilde{f}_{i,j}^{(1)} = 0 \,, \tag{8}$$

which shows that the sum of the first $(N-1)^2$ rows is zero. A similar property is observed for the second set of rows: their chequerboard combination is equal to zero. This condition requires

$$\sum_{i,j=1}^{N-1} (-1)^{i+j} \tilde{f}_{i,j}^{(2)} = 0 \,. \tag{9}$$

Finally we denote the solution vector by $w = (u \quad v)^T$. This is a column vector of length $2N^2 - 4N$ whose first $N^2 - 2N$ components represent the value of the solution u on the vertices of the mesh, and the remaining components represent the solution v, both ordered lexicographically. Hence, the problem (4), (5), (6) and (7) can be restated as

$$Aw = \tilde{f} \,. \tag{10}$$

Since this algebraic problem is overdetermined a solution can only be defined in a least squares sense, that is, by solving the *normal equations*

$$A^T A w = A^T \tilde{f} \,. \tag{11}$$

For the uniqueness of the solution the columns of the matrix A have to be linearly independent.

As an example, let us take $N = 3$, which is the coarsest grid to be used in a MG cycle (described later). Equations (4) and (5), together with the boundary conditions (6) and (7), provide 8 equations for 6 unknowns. In addition we have the compatibility conditions (8) and (9).

By solving the resulting system one obtains the solution values u and v, such that the residuals of all original equations are zero. However if the condition (9) is not satisfied, the least squares formulation still provides a unique solution to the problem

$$Aw = P_A \tilde{f} \,, \tag{12}$$

where

$$P_A = A(A^T A)^{-1} A^T \tag{13}$$

is the projection matrix onto the column space of A.

It is worth noting the relationship between the two cases when the conditions (8) and (9) are satisfied. In this case the least squares formulation is equivalent to the combination of the discrete divergence equations (defined on the same macro-cell, which is a square cell containing four cells)

by using the following pattern (where + or − means that the equation enters in the combination with a +1 or −1 multiplicative factor):

$$\begin{array}{cccccc} + & - & & - & - & & - & + \\ + & - & & + & + & & + & - \,. \end{array} \tag{14}$$

The discrete curl equations are combined as

$$\begin{array}{cccccc} + & - & & - & - & & + & + \\ + & - & & + & + & & + & + \,. \end{array} \tag{15}$$

The system of equations thus constructed coincides with $A^T A w = A^T \tilde{f}$. Notice that the idea of reducing the test space (of piecewise constant functions), described, for example in [14, 13], defines the same patterns depicted in (14) and (15).

In the remainder of this section we study P_A in detail. We find that, as we refine the mesh size h, P_A tends to the identity operator, as expected. This analysis is necessary to define the truncation error of the least squares equation (12). In fact the truncation error due to the cell vertex discretisation of the differential operators in (1) originates from the use of the trapezoidal rule

$$\int_a^b f(x)dx = \frac{1}{2}(f(a) + f(b))(b-a) - \frac{1}{12}(b-a)^3 f''(\tilde{x}) \,, \tag{16}$$

where $\tilde{x} \in (a, b)$. Thus in a cell of size h this summation introduces an error of order $O(h^3)$ on each side of the cell, and the global truncation will be a combination of all these contributions. In order to represent this error on each cell as a constant multiplied by some power of h, we derive each contribution by a Taylor expansion with respect to the center of the cell. This gives for the cell vertex Cauchy–Riemann equations a truncation error of order $O(h^2)$.

The analysis of P_A is necessary to define the approximation of the right-hand side of the discrete problem that we actually solve, with respect to that of the original problem (4) and (5). Fortunately it is possible to give P_A explicitly, in a compact way. It has the block structure

$$P_A = \begin{bmatrix} P_A^{(1)} & 0 \\ 0 & P_A^{(2)} \end{bmatrix}, \tag{17}$$

where each $P_A^{(i)}$ is idempotent and symmetric. Let us first consider $P_A^{(1)}$. For simplicity of notation denote by $q = \frac{1}{(N-1)^2}$. Thus, by using (8), we have

$$(P_A^{(1)} \tilde{f}^{(1)})_k = (\tilde{f}^{(1)})_k - q \sum_{i,j=1}^{N-1} \tilde{f}_{i,j}^{(1)} = (\tilde{f}^{(1)})_k \; ; \; k = 1, \ldots, (N-1)^2 \; ; \tag{18}$$

hence, because of the compatibility condition, $P_A^{(1)}$ acts on $\tilde{f}^{(1)}$ as an identity map.

It should be clear what to expect from $P_A^{(2)}$: when $\tilde{f}^{(2)}$ satisfies (9), $P_A^{(2)}$ acts as an identity operator on $\tilde{f}^{(2)}$. We have

$$(P_A^{(2)} \tilde{f}^{(2)})_k = (\tilde{f}^{(2)})_k - q(-1)^{(k+1-(N-1)^2)} \sum_{i,j=1}^{N-1} (-1)^{i+j} \tilde{f}_{i,j}^{(2)} \,, \tag{19}$$

$k = (N-1)^2 + 1, \ldots, 2(N-1)^2$.

Because, in principle, $\tilde{f}^{(2)}$ is not required to satisfy (9) we must evaluate the perturbation $(P_A^{(2)}\tilde{f}^{(2)} - \tilde{f}^{(2)})$. For this purpose, let us introduce the two-dimensional chequerboard function on Ω_h. Denoting the characteristic function on Ω_h^{ij} by χ_{ij}, the chequerboard function is

$$\chi_h(x,y) = \sum_{i,j=1}^{N-1} (-1)^{i+j}\chi_{ij} \ . \tag{20}$$

One can prove that χ_h weakly converges to zero (as $h \to 0$) in $L_2(\Omega)$ (see [13]).

To simplify the discussion which follows, without affecting the general validity of the result, we take $\Omega = (0,1)^2$ and assume homogeneous boundary conditions. Let us denote by $(\cdot,\cdot)$ the *Euclidean inner product*; then we can rewrite (19) as follows:

$$(P_A^{(2)}\tilde{f}^{(2)})_k = (\tilde{f}^{(2)})_k - h^2(\chi_h, \tilde{f}^{(2)})(\chi_h)_k \ , \ k = (N-1)^2 + 1, \ldots, 2(N-1)^2 \ . \tag{21}$$

We have the following theorem (for the proof see [1, 13]).

Theorem 1 *Suppose that* $\frac{\partial^2 f_2}{\partial x \partial y} \in L_2(\Omega)$, *and let* $N = 2^\ell + 1$, *with* ℓ *as some positive integer. Then*

$$|(\chi_h, \tilde{f}^{(2)})| \leq \frac{1}{3}\|\frac{\partial^2 f_2}{\partial x \partial y}\|_{L_2(\Omega)} \ . \tag{22}$$

Thus the perturbation $q\sum_{i,j=1}^{N-1}(-1)^{i+j}\tilde{f}_{i,j}^{(2)}$ due to the least squares approach is of order $O(h^2)$. By this approach we obtain a consistent algebraic problem which differs from the original one by an $O(h^2)$ perturbation of the right-hand side.

The stability and convergence analysis of the cell vertex approximation of the Cauchy–Riemann equations is presented in [13]. There we show that the cell vertex approximation is stable and second-order convergent in an appropriate H^1-norm. In particular the model problem which we also consider here is studied there as well. This gives an overdetermined system, for which the idea of reducing the test space is adopted, and stability and convergence properties follow. Similar convergence results are presented in [15] by using a least squares approach.

AN ITERATIVE SCHEME

From the discrete Cauchy Riemann equations it is clear that there is not a one-to-one correspondence between nodal values and equations. This means that a possible pointwise iteration must be constructed based on the cells, and thus it involves more than one cell. This is the case, for example, with the Lax–Wendroff iteration. Here we present a pointwise iteration procedure.

As in the previous section we start by analysing the simplest case ($N = 3$). This case actually appears in our computations, since it represents the coarsest problem in a multi-grid cycle. In the standard MG approach to solve simple model problems using, for example, a finite difference discretisation, the algebraic equations on the coarsest grid are solved exactly, and the 'solver' there coincides with one step of the iteration procedure (e.g., the pointwise Gauss–Seidel scheme). We now try to reproduce these aspects of an iteration for the cell vertex finite volume Cauchy–Riemann equations.

The variables which must be computed on the coarsest grid are represented as they appear on the grid:

$$\begin{array}{ccccc} \cdot & & u_{2,3} & & \cdot \\ v_{1,2} & u_{2,2}, & v_{2,2} & & v_{3,2} \\ \cdot & & u_{2,1} & \cdot & \cdot \end{array} \tag{23}$$

The iteration on the grid $N = 3$ must be capable of solving for all these variables in one step. Therefore the iteration is explicitly given by solving (11). The solution is given by

$$\begin{aligned} u_{2,1} &= (u_{1,2} + u_{3,2} + v_{1,1} - v_{3,1})/2 \\ &\quad +(\hat{f}^{(1)}_{1,1} - \hat{f}^{(1)}_{2,1} - \hat{f}^{(2)}_{1,1} - \hat{f}^{(2)}_{2,1})h/2 \,, \end{aligned} \tag{24}$$

$$\begin{aligned} u_{2,2} &= (u_{1,1} + u_{1,3} + u_{3,1} + u_{3,3})/4 \\ &\quad +(\hat{f}^{(1)}_{1,1} + \hat{f}^{(1)}_{1,2} - \hat{f}^{(1)}_{2,1} - \hat{f}^{(1)}_{2,2} \\ &\quad +\hat{f}^{(2)}_{1,1} - \hat{f}^{(2)}_{1,2} + \hat{f}^{(2)}_{2,1} - \hat{f}^{(2)}_{2,2})h/4 \,, \end{aligned} \tag{25}$$

$$\begin{aligned} u_{2,3} &= (u_{1,2} + u_{3,2} - v_{1,3} + v_{3,3})/2 \\ &\quad +(\hat{f}^{(1)}_{1,2} - \hat{f}^{(1)}_{2,2} + \hat{f}^{(2)}_{1,2} + \hat{f}^{(2)}_{2,2})h/2 \,, \end{aligned} \tag{26}$$

$$\begin{aligned} v_{1,2} &= (u_{1,1} - u_{1,3} + v_{2,1} + v_{2,3})/2 \\ &\quad +(\hat{f}^{(1)}_{1,1} - \hat{f}^{(1)}_{1,2} + \hat{f}^{(2)}_{1,1} + \hat{f}^{(2)}_{1,2})h/2 \,, \end{aligned} \tag{27}$$

$$\begin{aligned} v_{2,2} &= (v_{1,1} + v_{1,3} + v_{3,1} + v_{3,3})/4 \\ &\quad +(\hat{f}^{(1)}_{1,1} - \hat{f}^{(1)}_{1,2} + \hat{f}^{(1)}_{2,1} - \hat{f}^{(1)}_{2,2} \\ &\quad -\hat{f}^{(2)}_{1,1} - \hat{f}^{(2)}_{1,2} + \hat{f}^{(2)}_{2,1} + \hat{f}^{(2)}_{2,2})h/4 \,, \end{aligned} \tag{28}$$

$$\begin{aligned} v_{3,2} &= (-u_{3,1} + u_{3,3} + v_{2,1} + v_{2,3})/2 \\ &\quad +(\hat{f}^{(1)}_{2,1} - \hat{f}^{(1)}_{2,2} - \hat{f}^{(2)}_{2,1} - \hat{f}^{(2)}_{2,2})h/2 \,. \end{aligned} \tag{29}$$

Hence by substituting the values of the variables as given above, the coarsest problem is solved (in the least squares sense). Now we suppose that the mesh is refined by halving h. First, we notice that (24), (26), (27) and (29) provide the relaxation scheme for the boundary values, in the appropriate part of the domain's contour. The remaining two, (25) and (28), are suitable to relax the variables u and v in the interior of the domain. Note that they coincide with the pointwise Gauss–Seidel step for the Laplacian discretised with the usual skewed five point finite difference stencil. Actually, they reflect the fact that it is possible to combine the cell vertex Cauchy–Riemann equations (4) and (5) to obtain such a stencil.

So we obtain the Gauss–Seidel (GS) iteration for $A^T A w = A^T \tilde{f}$, also called *Kaczmarz* relaxation (see, e.g., [10]). Since $A^T A$ is a positive definite, symmetric matrix, the GS iteration converges to the solution, in the least squares sense, of the discrete Cauchy–Riemann equations. The analysis of the smoothing property of this iteration is carried out in [10, 11].

Remark 1 *After any iteration sweep the algebraic sum of the residuals for the divergence equations (4) is zero. Hence the compatibility condition for the corresponding residual equation for the solution error is satisfied. In fact, because of the (Dirichlet) boundary conditions, this error is zero on the boundary. This is an important property in order to solve the cell vertex Cauchy–Riemann equations by a multi-grid scheme.*

A FULL MULTI-GRID METHOD

Since we have the iterative scheme at hand we now need to define the restriction and prolongation operators to construct a multi-grid algorithm. The features of such operators are dictated, in part, by the problem we wish to solve and by the choice of the relaxation procedure. Even though our approach is based on least squares, we want a multi-grid code which in principle can be defined by using the properties of the discrete Cauchy–Riemann equations without involving the computation of the normal equations. A pure least squares approach would mean the development of an algebraic multi-grid method which applies to $A^T A w = A^T \tilde{f}$. We instead want an algorithm which works directly on $Aw = \tilde{f}$. The reason is that the resulting scheme can be easily adapted to more general problems. The first step was done in the previous section where we have defined a smoothing iteration. It has been obtained by solving the coarsest (least squares) problem.

On finer grids the solution of the normal equations is considered only locally. Clearly we had implicitly assumed to follow a standard MG approach. The approximation of the differential operator on any level is obtained by discretising it on that level. Let us consider, for the moment, the existence of two grids only, the coarse grid approximation of the differential operator is given by (4) and (5), but h is replaced by $H = 2h$. To distinguish the two problems defined on different grids we denote by A^h and A^H the cell vertex difference operators on Ω_h and Ω_H, respectively. In the same way, the vector functions w and $\tilde{f}$ on a given grid Ω_h will be denoted by w^h and $\tilde{f}^h$.

The definition of the transfer operator for the residuals is based on the way the finite volume method approximates the source terms. In the weak formulation provided by the cell vertex scheme, the source terms are discretised by integrating them on the given volume. Therefore the sum of the right hand sides of four discrete Cauchy–Riemann equations (of 'div' or 'curl' type) based on neighbouring cells Ω_h^{ij} with a common vertex provides the right hand side of a discrete Cauchy–Riemann equation based on the coarse cell Ω_H^{IJ}, which contains the fine cells. For this reason the transfer operator of the residuals from fine cells to a coarser one, is defined by the algebraic sum of the fine grid residuals contained in the coarse cell. That is, we have

$$(I_h^H r^h)_{IJ} = (r_{ij} + r_{i+1j} + r_{ij+1} + r_{i+1j+1})/4 \ , \tag{30}$$

where $_{i,j}$ and (I, J) refer to the same space point, and r_{ij} is the residual of the cell vertex Cauchy–Riemann equations (4) or (5) on the cell Ω_h^{ij}.

Also the definition of the prolongation operator follows in a natural way from the properties of the present discrete problem. Let us notice that, after a sufficient number of smoothing sweeps, $\|e\|_1 = (A^T A e, e) \approx 0$. Thus, from the algebraic multi-grid approach [11, 10], we can use the equation $A^T A e = 0$ and invert it to construct the prolongation operator. Therefore, by construction, the repeated application of the smoothing iteration produces an approximation whose error tends to lie in the range of the interpolation. By using $A^T A e = 0$ we obtain the interpolation formula for those fine points which are situated at the center of the coarse cell. The interpolated value of a variable on this grid point will be the mean value of those at the neighbouring vertices of the coarse cell containing the point. The remaining fine grid variables, on the edges of the coarse cells, are obtained by linear interpolation between the nearest two coarse variables. So we have I_H^h, the nine-point prolongation [16], symbolised by the stencil

$$\begin{bmatrix} 1/4 & 1/2 & 1/4 \\ 1/2 & 1 & 1/2 \\ 1/4 & 1/2 & 1/4 \end{bmatrix} . \tag{31}$$

It is interesting to notice the remarkable reduction properties of the transfer operators just described. It can be shown that

$$A^H = I_h^H A^h I_H^h , \tag{32}$$

which means that the coarse grid matrix problem that has been defined using the standard approach is obtainable from the fine grid matrix of coefficients and the given transfer operators, following a Galerkin approach. This property is very important. Since a sufficient number of relaxation sweeps produces an approximate solution w^h whose error e^h lies in the range of the prolongation operator, by using (32) the fine problem is reduced to a one of smaller size. The (least squares) solution of the coarse grid equation

$$A^H e^H = I_h^H (\tilde{f}^h - A^h w^h) \tag{33}$$

provides a good approximation for the fine grid error and is used in the coarse level correction $w^h := w^h + I_H^h e^H$.

A two level cycle is defined as the application of ν_1 pre-smoothing sweeps on the fine level, followed by the coarse level correction and ν_2 post-smoothing sweeps. If one uses the same method to determine e^H in (33) and the process is repeated recursively until the coarsest level is reached, then a multi-grid method is obtained.

At this stage there are some important points to be discussed. As one can notice, the least squares formulation is mainly used locally to develop a suitable smoothing iteration. The resulting relaxation scheme solves the normal equations. On the other hand, the remaining components of the multi-grid algorithm defined here are based on the original overdetermined Cauchy–Riemann equations. To make these points more clear we report in Figures 1 and 2 the values of the L_2 norm of the residuals of equations (4) and (5), and (11) as a function of work units, (i.e., the computation work invested to produce these residuals). In Figure 1 the simple relaxation is applied to solve the discrete problem on a given grid (this example is stated in the section of numerical experiments). The dotted line represents the residual norm of the cell vertex Cauchy–Riemann equations. The continuous line represents the residual of the normal equations. The same quantities are pictured in Figure 2 which reports the convergence history relative to the cyclic application of the MG scheme. The multi-grid method accelerates greatly the convergence of the Kaczmarz relaxation to the solution of the least squares problem. But the residual norm relative to the original Cauchy–Riemann equations converges to a non-zero value, since a solution for them does not exist. Because the relaxation and the coarse level correction are based on different equations, they are to some extent conflicting schemes. This fact appears in Figure 2 where for sufficiently small residuals which turn out to be of the order of the truncation error on the finest grid, the MG convergence slows down.

In order to define a full multi-grid scheme (see, e.g., [16]) we have to introduce another interpolation operator. We use a standard cubic interpolation operator. It is used to interpolate the solution to the problem on the level ℓ, after n multi-grid cycles, to the level $\ell + 1$, and so on recursively until the finest level M is reached and, finally, n multi-grid cycles are performed on level M. The resulting algorithm will be denoted by n-FMG. We shall test the n-FMG code described here in the section on numerical experiments. An equivalent scheme which solves the

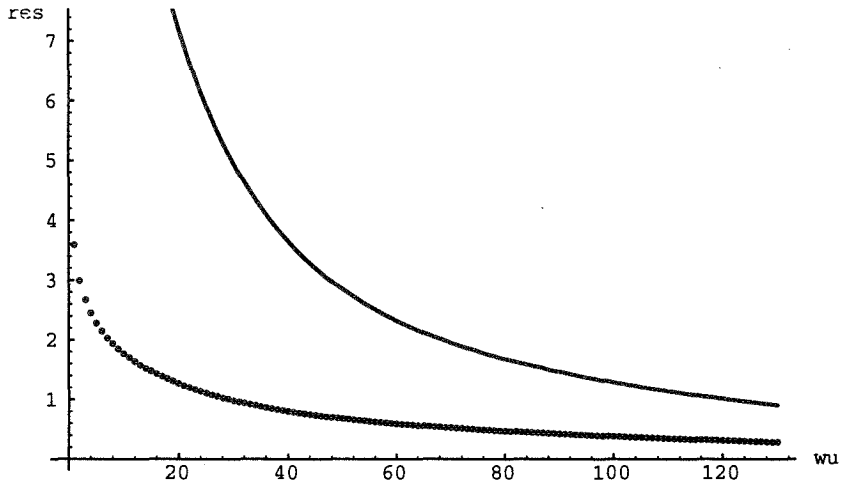


Figure 1: Convergence history for the L_2 residual norms relative to the cell vertex Cauchy–Riemann equations (dotted line) and the normal equations (continuous line) when the Kaczmarz relaxation is applied (for the u component).

cell vertex Cauchy–Riemann equations based on triangles has also been tested, giving similar results [1].

NUMERICAL EXPERIMENTS

In this section we report the results of some numerical experiments. As we have previously seen, the multi-grid iteration has a convergence rate which slows down after a large number of iterations. However an optimal multi-grid method results when the MG cycle is used in combination with a nested iteration technique, thus resulting in a full multi-grid scheme. In fact we show that the full multi-grid scheme previously described is capable of solving the discrete Cauchy–Riemann equations to the level of the truncation error $O(h^2)$ employing only one MG cycle at each current level of discretisation. We consider the Cauchy–Riemann equations discretised on the square $(0,2) \times (0,2)$. Some numerical parameters are fixed, namely, the coarsest mesh size $h_1 = 1$; the number of intervals of the coarsest grid equals 2 in both directions. The initial starting approximation is always the zero function (except on the boundary).

The first example has been previously considered to obtain Figures 1 and 2 (employing five levels); the source terms are (integrated over $[x, x+h] \times [y, y+h]$) given by

$$\begin{aligned}\hat{f}^{(1)}(x,y) = &-\tfrac{1}{h^2}((a-b)(\cos(by) - \cos(b(h+y)))\sin(ax)/(ab) \\ &+(a-b)(\cos(by) - \cos(b(h+y)))\sin(a(h+x))/(ab))\,,\end{aligned} \tag{34}$$

$$\begin{aligned}\hat{f}^{(2)}(x,y) = &-\tfrac{1}{h^2}((a+b)\cos(ax)(\sin(by) - \sin(b(h+y)))/(ab) \\ &+(a+b)\cos(a(h+x))(\sin(by) - \sin(b(h+y)))/(ab))\,.\end{aligned} \tag{35}$$

The exact solution and the boundary values are given by

$$u(x,y) = \sin(ax)\sin(by)\,, \tag{36}$$

$$v(x,y) = \cos(ax)\cos(by)\,, \tag{37}$$

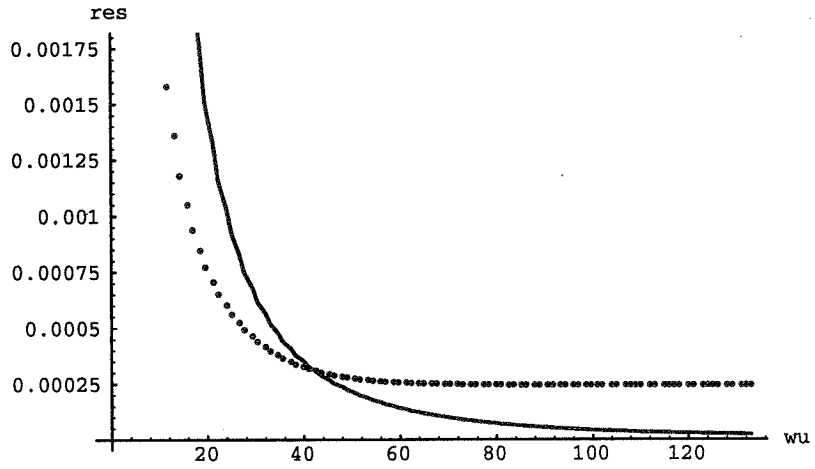


Figure 2: Convergence history for the L_2 residual norms relative to the cell vertex Cauchy–Riemann equations (dotted line) and the normal equations (continuous line) when the multi-grid scheme is used (for the u component).

where a and b are given parameters. The behaviours reported in Figures 1 and 2 correspond to the case in which $a = 1$ and $b = 2$. Now let us consider $a = b = 1$,. so that $\hat{f}^{(1)} = 0$.

We have seen that the multi-grid step has satisfactory convergence properties in the first few iterations whenever the initial approximation produces residuals of the cell vertex Cauchy–Riemann equations which are larger than those corresponding to the exact solution of the least squares problem. Therefore it is convenient to use the MG code to work within this limit, which suffices in order to obtain an efficient FMG algorithm.

Table 1: The Behaviour of the L^2 Norm of the Solution Error for Various n-FMG, for the u and v Components.

	u		v	
M	1-FMG	2-FMG	1-FMG	2-FMG
2	0.19(-1)	0.17(-1)	0.12(-1)	0.77(-2)
3	0.36(-2)	0.32(-2)	0.23(-2)	0.20(-2)
4	0.81(-3)	0.71(-3)	0.57(-3)	0.53(-3)
5	0.19(-3)	0.17(-3)	0.15(-3)	0.14(-3)
6	0.46(-4)	0.40(-4)	0.37(-4)	0.34(-4)
WU	3.5	7.1	3.5	7.1

Table clearly shows† that n-FMG with $n = 1$ is sufficient to solve the problem to the order of the truncation error. The work invested in the FMG process is measured in *work units* (WU), that is, the computational work of one relaxation sweep on the finest grid.

†In this table the power 10^{-k} is represented by $(-k)$. We refer to the usual discrete L^2 norm of the error between the numerical and the analytical solution.

The same experiment is then repeated with different values of a and b. We always observe the behaviour described above and quadratic convergence of the numerical solution.

So far we have considered structured uniform grids. We now comment on how to generalise the present algorithm to solve the Cauchy–Riemann equations on non-uniform quadrilateral grids. For this purpose we consider the generalised Lax–Wendroff (LW) scheme (see [17, 12]). This technique is based on time-stepping the (artificial) unsteady problem, derived from the original one by adding a partial derivative in time of the unknown variables. Then, on each node, a new value for the solution vector is obtained from the previous one by a Taylor series in time, up to second order terms. The first and the second order term are then discretised by applying cell vertex finite volume techniques based on the macro-cell [12].

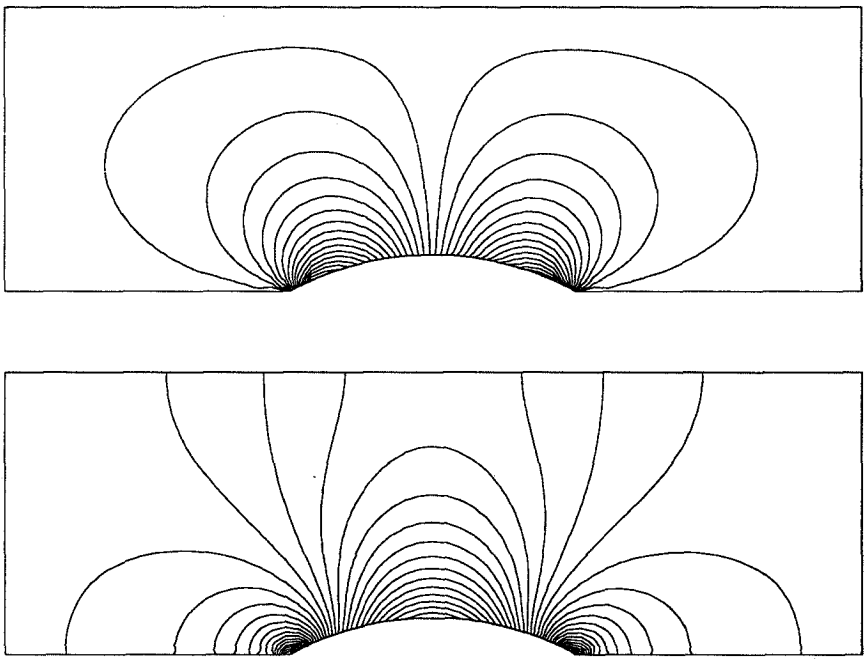


Figure 3: Plots of contour lines of the functions v (top) and u (bottom).

Now we can prove that, for uniform quadrilateral grids, the cell vertex approximation of the second order term in the Lax–Wendroff method is given (up to the multiplicative constant δt^2) by

$$-\frac{1}{2}A^T(Aw - \tilde{f})\,. \tag{38}$$

Hence, the application of a LW iteration, which consists of only the second order term (take $\delta t = h$), is equivalent to the Kaczmarz relaxation. On the other hand we notice that once the grid is non-uniform the least squares approach is difficult to apply, while the corresponding Lax–Wendroff iteration is of immediate application. Therefore, on non-uniform grids we extend the full multi-grid algorithm presented above by using a second order LW iteration as a smoother. As a simple example of application we use this algorithm to solve the homogeneous Cauchy–Riemann equations on the geometry of the "bump" problem, subject to the condition $(u, v)_n = 0$, except at the inflow and outflow boundaries where $u = 1$. In Figure 3 we plot the

contour line of the function u computed by a 3-FMG method ($M = 6$).

ACKNOWLEDGMENTS

The first author wishes to thank Nick Birkett for many helpful discussions and for the development of the grid generation program for the "bump" problem.

REFERENCES

[1] A. Borzì, K.W. Morton, E. Süli, and M. Vanmaele. Multilevel Solution of Cell Vertex Cauchy-Riemann Equations. Technical Report NA94/8, Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford OX1 3QD, 1994.

[2] S. Ta'asan. Canonical Forms of Multidimensional Steady Inviscid Flows. ICASE Technical Report 93-34, Institute for Computer Applications in Science and Engineering, Hampton, Virginia 23681-0001, 1993.

[3] A. Brandt and N. Dinar. Multigrid Solutions to Elliptic Flow Problems. In S.V. Parter, editor, *Numerical Methods for Partial Differential Equations.* Academic Press, New York, 1979.

[4] A. Brandt and I. Yavneh. On multigrid solution of high-Reynolds incompressible entering flows. *Journal of Computational Physics*, 101:151–164, 1992.

[5] A. Brandt and I. Yavneh. Accelerated multigrid convergence and high-Reynolds recirculating flows. *SIAM Journal of Scientific Computation*, 14:607–626, 1993.

[6] S. Ta'asan. Canonical-Variables Multigrid Method for Steady-State Euler Equations. ICASE Technical Report 94-14, Institute for Computer Applications in Science and Engineering, Hampton, Virginia 23681-0001, 1994.

[7] M.E. Rose. A 'unified' numerical treatment of the wave equation and the Cauchy-Riemann equations. *SIAM Journal of Numerical Analysis*, 18(2):372–376, 1981.

[8] G.J. Fix and M.E. Rose. A comparative study of finite element and finite difference methods for Cauchy-Riemann type equations. *SIAM Journal of Numerical Analysis*, 22(2):250–261, 1985.

[9] T.B. Gatski, C.E. Grosch, and M.E. Rose. A Numerical Study of the Two-Dimensional Navier-Stokes Equations in Vorticity-Velocity Variables. *Journal of Computational Physics*, 48:1–22, 1982.

[10] A. Brandt. Algebraic Multigrid Theory: The Symmetric Case. *Applied Mathematics and Computation*, 19:23–56, 1986.

[11] J.W. Ruge and K. Stüben. Algebraic Multigrid. In S.F. McCormick, editor, *Multigrid Methods.* SIAM, Philadelphia, 1987.

[12] P.I. Crumpton, J.A. Mackenzie, and K.W. Morton. Cell Vertex Algorithms for the Compressible Navier-Stokes Equations. *Journal of Computational Physics*, 109:1–15, 1992.

[13] M. Vanmaele, K.W. Morton, E. Süli, and A. Borzì. Analysis of the cell vertex finite volume method for the Cauchy-Riemann equations. Technical Report NA94/5, Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford OX1 3QD, 1994.

[14] C. Johnson and J. Pitkäranta. Analysis of some mixed finite element methods related to reduced integration. *Mathematics of Computation*, 38(158):375–400, 1982.

[15] C-L. Chang and M.D. Gunzburger. A subdomain-Galerkin/least squares method for first-order elliptic systems in the plane. *SIAM Journal of Numerical Analysis*, 27(5):1197–1211, 1990.

[16] W. Hackbusch. *Multi-Grid Methods and Applications.* Springer-Verlag, Heidelberg, 1985.

[17] K.W. Morton, M.A. Rudgyard, and G.J. Shaw. Upwind Iteration Methods for the Cell Vertex Scheme in One Dimension. *Journal of Computational Physics*, 114:209–226, 1994.

MULTILEVEL ALGORITHM FOR ATMOSPHERIC DATA ASSIMILATION*†

Achi Brandt and Leonid Yu. Zaslavsky
Department of Applied Mathematics and Computer Science
The Weizmann Institute of Science
Rehovot, 76100, Israel

SUMMARY

A multiscale algorithm for the problem of optimal statistical interpolation of observed data has been developed. This problem includes the calculation of the vector of the "analyzed" (best estimated) atmosphere flow field w^a by the formula

$$w^a = w^f + P^f H^T y,$$

where the quantity y is defined by the equation

$$(HP^f H^T + R)y = w^o - Hw^f,$$

using the given model forecast first guess w^f and the vector of observations w^o; H is an interpolation operator from the regular grid to the observation network, P^f is the forecast error covariance matrix, and R is the observation error covariance matrix.

At this initial stage the case of univariate analysis of single level radiosonde height data is considered. The matrix R is assumed to be diagonal, and the matrix P^f is assumed to be given by the formula $P^f_{ij} = \sigma^f_i \mu_{ij} \sigma^f_j$, where μ_{ij} is a smooth, decreasing function of the distance between the ith and the jth points.

Two different multiscale constructions can be used to efficiently solve the problem of optimal statistical interpolation: a technique for fast evaluation of the discrete integral transform $\sum_i P^f_{ij} v_j$, and a fast iterative process which effectively works with a sequence of spatial scales. In this paper we describe a multiscale iterative process based on a multiresolution, simultaneous displacement technique and a localized variational calculation of iteration parameters.

*A preliminary version of the material presented here has been presented in [1].

†The second author has been supported by a Sir Charles Clore Post-Doctoral Fellowship.

E-mail addresses: mabrandt@weizmann.weizmann.ac.il and zasl@wisdom.weizmann.ac.il

INTRODUCTION

The problem of optimal statistical interpolation of the observed data includes the calculation of the vector of the "analyzed" (best estimated) atmosphere flow field w^a by the formula

$$w^a = w^f + P^f H^T y,$$

where the quantity y is determined from the equation

$$(HP^f H^T + R)y = w^o - Hw^f, \tag{1}$$

using the given model forecast first guess w^f and the vector of observations w^o ([2]–[4]). Typically, w^f is defined on a regular spherical grid, while the set of observations w^o is defined on an irregular network of observation points; H is an interpolation operator from the regular grid to the observation network, P^f is the forecast error covariance matrix, and R is the observation error covariance matrix.

The observation error covariance matrix R is assumed to be diagonal with

$$R_{ii} = (\sigma_i^o)^2.$$

The forecast error covariance function $P^f(x_1, x_2)$ is defined for any pair of points x_1 and x_2 on the sphere by the formula

$$P^f(x, z) = \sigma^f(x)\mu(x, z)\sigma^f(z),$$

where the forecast error correlation function $\mu(x, z)$ is described as a smooth, decreasing function of the distance between the points x and z [4]. The matrices P^f and μ are the restrictions of functions $P^f(x, z)$ and $\mu(x, z)$ on the regular latitude-longitude grid.

The purpose of this paper is to conceptualize a fast multiscale iterative process for solving y from equation 1 when the observation network is strongly inhomogeneous in space. At this initial stage, we consider a univariate analysis of single level radiosonde height data.

In this paper we consider only convergence properties of the iterative process. Accordingly, in the computer experiments all summations have been performed in a straightforward manner. An effective procedure for the fast evaluation of the integral transform on the sphere, based on the Brandt and Lubrecht approach [5], will be presented separately.

Without loss of generality, equation (1) can be replaced by the system of equations

$$\sum_j \tilde{P}_{ij}^f y_j + R_{ii} y_i = w_i^o - (Hw^f)_i, \tag{2}$$

where

$$\tilde{P}_{ij}^f = \tilde{\sigma}_i^f \mu_{ij} \tilde{\sigma}_j^f$$

for the ith and jth observation points x_i and x_j,

$$\mu_{ij} = \mu(x_i, x_j),$$

and

$$\tilde{\sigma}_i^f = (H\sigma^f)_i.$$

(While the matrix P is defined for the points of the regular grid and interpolated to the observation network using the operator H, the matrix $\tilde{P}$ is defined by the same formula directly on the observation network.)

Indeed, the difference

$$HPH^T y - \tilde{P}y$$

may be treated as an additional source on the right hand side. This small term is nonprincipal at all scales and can easily be taken into account in iterations.

Since we want to deal explicitly with the smoothness properties of the kernel μ_{ij}, we replace (2) by

$$\sum_j \mu_{ij} u_j + (\frac{\sigma_i^o}{\tilde{\sigma}_i^f})^2 u_i = \frac{1}{\tilde{\sigma}^f}(w_i^o - (Hw^f)_i), \tag{3}$$

where $u_i = \tilde{\sigma}_i^f y_i$. The system of equations (3) can be written in matrix notation as

$$Au = f, \tag{4}$$

where the matrix A is symmetric and presumably positive definite.

GENERAL STRATEGY

It is important to understand why many common iterative processes, such as Jacobi, Gauss-Seidel, or conjugate gradient, converge slowly when applied to equation (4). Let us consider, for example, the simplest iterative process

$$u^{(n+1)} = u^{(n)} + \omega r^{(n)}, \tag{5}$$

where the residual

$$r^{(n)} = f - Au^{(n)},$$

and parameter $\omega \approx (\rho(A))^{-1}$, where $\rho(A)$ is the spectral radius of the operator A.

The process (5) reduces effectively the error components that correspond to the large eigenvalues λ_l, such that

$$\omega\lambda_l \sim 1,$$

while the error components that correspond to the small eigenvalues λ_s, for which

$$\omega\lambda_s \ll 1,$$

are reduced slowly [6]. Since the summation $\sum_j \mu_{ij} u_j$ in (3) is made with a smooth kernel, eigenvectors of A that correspond to the large eigenvalues are (mostly) spatially

smooth, and eigenvectors of A that correspond to small eigenvalues are oscillatory in space. Therefore, one cannot define one particular value of ω that would give an essential reduction of all spectral error components.

The effect described above is well-studied for the case when (4) is obtained as a grid approximation of the continuous integral equation. A few multiscale techniques based on multigrid ([5],[7]) and wavelet [8] approaches were developed in the 1990's. Unfortunately, these techniques cannot be applied to the considered problem in a straightforward manner because of the strong inhomogeneity of the observation

network.

The central idea of the approach developed below is to filter sequentially spectral components of $r^{(n)}$ and to choose for each a value of the iteration parameter that gives an essential reduction of the corresponding error component.

The major particular difficulty which has been overcome successfully in this work is how to define variable pass spatial filters $\mathcal{F}_h$, depending on the scale parameter h, for a field defined on a very inhomogeneous network. An appropriate filter will be described in §3.

When some component $\mathcal{F}_h r^{(n)}$ of the residual $r^{(n)}$ has been filtered, one should next calculate the correction vector. A simple way to do it is to use a *scalar* iteration parameter ω_h (i.e., to calculate the correction as $\omega_h \mathcal{F}_h r^{(n)}$). Then the modified iterative process (5) can be written as

$$u^{(n+1)} = u^{(n)} + \omega_h^{(n)} \mathcal{F}_h r^{(n)}, \tag{6}$$

where the iteration parameter depends on the scale h in some way.

An intrinsic disadvantage of schemes like (6) is that one *global* iteration parameter $\omega_h^{(n)}$ is determined for the entire domain. The optimal correction at a spatial point x_i should, however, depend only on the residual values at points located at most a few h from x_i. Therefore, in §4, we construct a procedure for calculating an iteration parameter $\omega_{h,i}^{(n)}$ for each point x_i *locally*, using only the values of the residual components in some area around x_i. This means that the iterative process which we construct can be written as

$$u^{(n+1)} = u^{(n)} + \Omega_h^{(n)} \mathcal{F}_h r^{(n)}, \tag{7}$$

where

$$\Omega_h = \mathrm{diag}(\omega_{h,i}^{(n)}).$$

We discuss the structure of the multilevel iterative cycle in §5.

SPATIAL FILTER

We now define a filter applicable to functions defined on a very irregular discrete network. Obviously, we want our filter to work like a usual spectral high pass filter in the data dense regions.

What do we want to get in regions of sparse data? Suppose we have an observation point s which is separated from other points by distance

$$d_s = \min_{p \neq s} \mathrm{dist}(s, p).$$

We would like to take into account the residual component r_s only on the scales h that are large enough:

$$h \simeq d_s \text{ and } h > d_s;$$

we neglect r_s on the small scales h:

$$h \ll d_s.$$

We define the filter $\mathcal{F}_h$ which satisfies these requirements by the formula

$$(\mathcal{F}_h r)_i = r_i - \gamma_i \sum_j r_j \exp\left(-\frac{1}{2}\frac{\mathrm{dist}^2(i,j)}{h^2}\right), \tag{8}$$

where h is the current scale, and the parameter γ_i is defined by the formula

$$\gamma_i = \left(\sum_j \exp\left(-\frac{1}{2}\frac{\mathrm{dist}^2(i,j)}{h^2}\right)\right)^{-1}.$$

Note that the filter can be calculated efficiently using the fast summation procedure.

CALCULATION OF Ω_h

The *scalar* iteration parameter $\omega_h^{(n)}$ in (6) can be determined from the variational condition of minimizing the Euclidean norm of the scale h component of the new residual $r_h^{(n+1)}$:

$$\min_{\omega_h^{(n)}} ||\mathcal{F}_h(r^{(n)} - \omega_h^{(n)} A \mathcal{F}_h r^{(n)})||^2,$$

where $||\cdot||$ is the Euclidean norm on the observation network

$$||u||^2 = (u, u)$$

and

$$(u, v) = \sum_i u_i v_i,$$

where the summation is made over the observation points. This condition leads to the formula

$$\omega_h^{(n)} = \frac{(p, q)}{(q, q)}, \tag{9}$$

where

$$p = \mathcal{F}_h r^{(n)},$$
$$q = \mathcal{F}_h A p.$$

As mentioned above, one disadvantage of this formula is its globality. In order to localize it, we use a family of weighted Euclidean norms. Let us introduce

$$(u,v)_{l,i} = \sum_j u_j v_j \exp\left(-\frac{1}{2}\frac{\mathrm{dist}^2(i,j)}{l^2}\right),$$

where the summation is made over the observation points and

$$||u||^2_{l,i} = (u,u)_{l,i}.$$

Now we can define the matrix Ω_h in (7). We choose Ω_h as follows:

$$\Omega_h^{(n)} = \mathrm{diag}(\omega_{h,i}^{(n)}),$$

where

$$\omega_{h,i}^{(n)} = \frac{(p^{(n)}, q^{(n)})_{3h,i}}{(q^{(n)}, q^{(n)})_{3h,i}},$$
$$p^{(n)} = \mathcal{F}_h r^{(n)},$$
$$q^{(n)} = \mathcal{F}_h A p^{(n)}.$$

Note again that the fast summation procedure can be used to calculate $\omega_{h,i}^{(n)}$ efficiently.

STRUCTURE OF THE MULTISCALE ITERATIVE CYCLE

In order to define the order of the multiresolution iterations, we have to prescribe some spatial scale to each iteration. The current spatial scale is determined by the formula

$$h = H \cdot 2^{1-level(n)},$$

where H is the largest scale and $level(n)$ is the level prescribed for the nth iteration. If $level(n) = 0$, the filter is not used. The multiscale iterative algorithm can be written as follows:

DO n = 1, NITER

Residual calculation

$r^{(n)} = f - Au^{(n)}$

IF $level(n) > 0$ THEN

Definition of the current scale

$h = H \cdot 2^{1-level(n)}$

Filtering

$p^{(n)} = \mathcal{F}_h r^{(n)}$

$q^{(n)} = \mathcal{F}_h A p^{(n)}$

Calculation of the iteration parameters

$\omega_{h,i}^{(n)} = \frac{(p^{(n)},q^{(n)})_{3h,i}}{(q^{(n)},q^{(n)})_{3h,i}}$

$\Omega_h^{(n)} = \mathrm{diag}(\omega_{h,i}^{(n)})$

Calculation of the new approximation to the solution

$u^{(n+1)} = u^{(n)} + \Omega_h^{(n)} p^{(n)}$

ELSE

Filtering is not used

$p^{(n)} = r^{(n)}$

$q^{(n)} = A p^{(n)}$

Calculation of the iteration parameter

$\omega_h^{(n)} = \frac{(p^{(n)},q^{(n)})}{(q^{(n)},q^{(n)})}$

Calculation of the new approximation to the solution

$u^{(n+1)} = u^{(n)} + \omega_h^{(n)} p^{(n)}$

ENDIF

ENDDO

We have used for our initial tests the standard V(2,2) multilevel cycle with 3 iterations at the 0th level ([9], [10]). This means that the function $level(n)$ is periodic:

$$level(LC + k) = level(k) \quad \text{for any } k > 0,$$

and

$$level(n) = \begin{cases} NLVL - k + 1, & \text{if } n = 2 \cdot k - 1 + l;\ k = 1, 2, \ldots, NLVL;\ l = 0, 1 \\ 0, & \text{if } n = 2 \cdot NLVL + l;\ l = 1, 2, 3 \\ k, & \text{if } n = 2 \cdot NLVL + 2 + 2k + l;\ k = 1, 2; \ldots, NLVL;\ l = 0, 1 \end{cases}$$

where $NLVL$ is the finest level number and $LC = 4 \cdot NLVL + 3$.

NUMERICAL RESULTS

At this initial stage of the work we made all the numerical tests with radiosonde height data only. The forecast correlation function is modeled by the formula

$$\mu(x_1, x_2) = \left(1 + \frac{(\text{dist}(x_1, x_2))^2}{L^2}\right)^{-1.208}$$

where $\text{dist}(x_1, x_2)$ is the three-dimensional distance between points x_1 and x_2 and L is the correlation distance ($L = 951$ km).

The radiosonde station locations and values of σ^o, σ^f, and $w^o - Hw^f$ were obtained from the Data Assimilation Office of NASA/Goddard Space Flight Center. The data file contains model parameters and radiosonde height observations from 715 stations. Observation error variances σ_i^o were taken to be equal to 14.6 m for all radiosonde stations. Forecast error variances σ_i^f vary from point to point and range from 18 m to 35 m.

We made our experiments with $NLVL = 5$. The scales which were used are shown in Table 1. The results of our experiments are shown in Table 2.

Table 1. Scale structure

Level number	1	2	3	4	5
Scale h, km	10 000	5 000	2 500	1 250	625

Table 2. Convergence of the iterative procedure

Multiscale cycle	L_2 norm of the residual	Rate of decrease of the norm
Initial	2.5510^{+1}	
1	6.8610^{-1}	0.027
2	3.7610^{-2}	0.054
3	2.0510^{-3}	0.054
4	8.8710^{-5}	0.043

DISCUSSION: FURTHER IMPROVEMENTS

The algorithm described above represents the first step toward development of a fast and efficient solver for the atmospheric data assimilation problem. It has been shown that the multiresolution algorithm can provide a fast solver. As long as the number of measurements is moderate (e.g., less than a few thousand), this algorithm

by itself is already effective enough. However, for larger sets of measurements, a major part of the work per cycle can be saved by a more advanced multiscale algorithm that features the following improvements.

First, as already mentioned, a fast evaluation of the operator P^f (i.e., of the multi-summation in Eq. (3)), can be based on the method of [5]. (See also [7]). Fast multi-summation can also be used for fast filtering.

Secondly, at each scale h in any region where the number of of measurements per $O(h \times h)$ cell is large, the multitude of residuals can be replaced by their proper local averages on a grid with mesh size $O(h)$. Similarly in such regions, the correction $u^{(n+1)} - u^{(n)}$ will also be calculated on such a grid and will only later be interpolated to the measurement points. Actually, the residual averagings and the correction interpolation will not be done directly between the finest (measurements) level and each scale-h level but will be transferred sequentially through all intermediate levels.

Thirdly, the residual filtering can be replaced by distributive relaxation (as in [5]). The latter is simple in the grid regions mentioned above. In other regions, the filtering techniques may be easier to apply.

These improvements will reduce the work of a cycle to a few operations per measurement, hopefully retaining the same convergence rates.

ACKNOWLEDGMENTS

This paper presents part of a joint effort with Drs. Steve Cohn, Greg Gaspari, and Arlindo da Silva of the Data Assimilation Office, NASA Goddard Space Flight Center. The authors thank them for raising interest in the problem of atmospheric data assimilation and for providing the observation data and the model parameters.

The authors are indebted to Drs. Alexander Kheifets, Ilya Rivin, Mikhail Shapiro, and Kees Venner for helpful discussions.

REFERENCES

[1] Brandt, A. and L. Yu. Zaslavsky, 1995: Multiscale Algorithm for Atmospheric Data Assimilation. Part I. Multiscale Iterative Process. *Technical Report CS95-09.* The Weizmann Institute of Science, Rehovot, Israel, pp. 9.

[2] Daley, R., 1991: *Atmospheric Data Analysis.* Cambridge University Press, pp. 457.

[3] Baker, W. E., S. C. Bloom, J. S. Woollen, M. S. Nestler, E. Brin, T. W. Schlatter, and G. W. Branstator, 1987: Experiments with a three-dimensional statistical objective analysis scheme using FGGE data. *Mon. Wea. Rev.*, **115**, 273-296.

[4] Pfaendter, J., S. Bloom, D. Lamich, M. Seabloom, M. Sienkiewicz, J. Stobie, and A. da Silva, 1995: Documentation on the Goddard Earth Observing System (GEOS) Data Assimilation System - Version 1, NASA TM-104606, Vol. 4, NASA, Goddard Space Flight Center, Greenbelt, MD.

[5] Brandt, A. and A. A. Lubrecht, 1990: Multilevel Multi-Integration and Fast Solution of Integral Equations. *J. Comput. Phys.*, **90**, 348-370.

[6] Varga, R. S., 1962: *Matrix Iterative Analysis.* Prentice-Hall, Inc., NJ, pp. 322.

[7] Brandt, A., 1991: Multilevel algorithms of integral transforms and particle integrations with oscillatory kernels. *Comp. Phys. Commun.*, **65**, 24-38.

[8] Alpert, B., G. Beylkin, R. Coifman, and V. Rokhlin, 1993: Wavelet-like bases for the fast solution of second-kind integral equations. *SIAM J. Sci. Comput.*, **14**, 159-184.

[9] Brandt, A., 1977: Multi-level adaptive solutions to boundary-value problems. *Math. Comp.*, **31**, 333-390.

[10] Hackbusch, W., 1985: *Multi-Grid Methods and Applications.* Springer Verlag, Berlin, pp. 377.

Effective Boundary Treatment for the Biharmonic Dirichlet Problem*

A. Brandt

J. Dym†

Department of Applied Mathematics and Computer Science
The Weizmann Institute
Rehovot, 76100, Israel

SUMMARY

The biharmonic equation can be rewritten as a system of two Poisson equations [6, 4]. Multigrid solution of this system is expected to converge with the same amount of work as solving two Poisson equations, requiring less than 70 floating point operations (scalar multiply or addition) per fine grid point to reach a solution using an FMG algorithm. For periodic boundary conditions, this goal is attained by simple, straightforward application of multigrid. For Dirichlet boundary conditions, however, convergence is impeded by poor interaction with the boundaries. Attempts to overcome the slowness without specifically addressing the boundaries have resulted in multigrid algorithms not attaining the Poisson convergence rate [3, 7].

We present three methods of boundary treatment with which full multigrid efficiency can be obtained. All implement an approach described by Brandt [1], concentrating some additional effort near the boundary. The first approach [9, 5] simply adds a number of relaxation sweeps over points close to the boundary. The second [8] uses joint relaxation on near-boundary points. The third method [5] takes something from each of the first two methods, resulting in a solver more suitable for highly parallel applications.

*Research Supported by Israel Ministry of Science Grant 4135-1-93, and by the C. F. Gauss Minerva Center for Scientific Computation at the Weizmann Institute of Science, Rehovot, Israel.

†Present Address: Department of Mathematics DRB 155, USC, 1042 W. 36th Pl, LA, Calif., 90089-1113.

Introduction

The biharmonic operator surfaces in a large number of applications. It is more efficiently solved as a system of two Poisson equations. The finite difference multigrid solver for this system is sensitive to the boundary conditions associated with the problem; for some, fast convergence is achieved with no special effort, while other boundary conditions require careful treatment of the gridpoints at and about the boundary to attain full multigrid efficiency.

The Dirichlet boundary conditions are an example of the second type. Without special care, the multigrid boundary convergence rate dominates the process after a short while, slowing down the entire process. Several methods for treating the boundary have been developed. Two are presented here, along with a newly devised method, more suitable for use with parallel computation.

The Biharmonic Dirichlet Problem

The biharmonic equation is

$$\triangle^2 u = f \tag{1}$$

within a given domain Ω, along with two boundary conditions on $\partial\Omega$, where $\triangle$ represents the Laplacian operator. The Dirichlet boundary conditions are

$$\begin{aligned} u &= \phi \\ \frac{\partial u}{\partial n} &= \psi \end{aligned} \tag{2}$$

on the boundary $\partial\Omega$. In [4], Ciarlet and Raviart quote Glowinski [6], who suggested that the equation can be more efficiently solved as a system,

$$\begin{aligned} \triangle u - v &= 0 \\ \triangle v &= f. \end{aligned} \tag{3}$$

They prove that this system of equations can be solved with (single-level) efficiency equal to that of a Poisson equation solver[1]. The problem has also been extensively analyzed in multigrid literature [8, 9, 3, 7]. The latter two prove formally the convergence of straightforward multigrid solvers for (3), or slight modifications thereof (in [7]). The algorithms considered use a relaxation sweep over the entire domain as the smallest 'unit of computation'. As a result, they are slow, requiring a large number of sweeps at each level to converge [3] or converging relatively slowly [7]. The algorithms described in the former two references implement (in different ways) an idea described by Brandt [1], concentrating on the area near the boundary, where slow convergence holds up the entire process.

[1]Note that evaluating both equations of the system requires less work than evaluating the biharmonic equation, although this, of course, is not the main computational benefit.

Basis For Comparison

The desired cycling convergence rate for multigrid solvers of (3) is the interior smoothing rate of the Poisson equation (for whatever number of relaxations is performed on the finest grid per cycle). For lexicographic ordering, this amounts to a factor of 2 per sweep, while for Red-Black ordering it is 4 for one sweep, 16 for two and 27 for three [1].

These rates, however, are achievable only for 2-level algorithms with ideal intergrid transfers. For practical multigrid applications, the convergence rate can be somewhat smaller, depending on the cycle parameters (including choice of intergrid transfers). Thus, the convergence of the Dirichlet boundary condition solver should be compared with that of a solver of (3) with *periodic* boundary conditions and otherwise identical parameters (there being no coupling of the equations near the boundary, this behaves exactly like two Poisson equations; note that an additional constraint must be supplied to each equation to make it nonsingular). Table 1 shows the convergence rates attained for various types of cycles, using full weighting and bi-linear interpolation for coarsening and prolongation respectively and Red-Black relaxation ordering. Using higher order interpolation would reduce the gap between the $W(2,1)$ smoothing rate and attained periodic convergence.

Cycle	$W(1,1)$	$W(2,1)$	$V(1,1)$	$V(2,1)$
Interior Smoothing Rate	16	27	16	27
Periodic B.C. convergence	15	19	8.5	12.5

Table 1: Comparison basis. Interior smoothing rates and attained multigrid cycling rates for periodic boundary conditions.

Boundary Condition Discretization

The Dirichlet boundary conditions (2) constrain the values of u on the boundary and of the derivative of u normal to the boundary to some given values. The normal derivative is discretized by introducing a set of virtual u-points, parallel to the boundary and one mesh size from it (*e.g.*, for the unit square, lines of points $u(-h,y)$, $u(1+h,y)$, $u(x,-h)$, and $u(x,1+h)$). Typically, the normal derivative is approximated using a central difference between the virtual points and the first interior u-points.

The second equation in (3) holds in the interior, but not on the boundary or the virtual layer. v, however, is defined on the boundary as well as in the interior. When performing relaxation, the boundary conditions on u and on $\frac{\partial u}{\partial n}$ are used to set u values on the true and virtual boundary[2], respectively, while the first equation in (3) on the boundary is used to determine v there.

[2]Actually, in all the algorithms implemented here, the virtual layer is implicit. The normal derivative boundary condition is used to substitute interior and boundary values of u whenever a virtual u is called for.

Boundary Slowdown

Applying the multigrid algorithm to the Dirichlet problem results in convergence rates much slower than those obtained with periodic boundary conditions. Figure 1 describes the problem better than a hundred words (but we'll try anyway...). Clearly, the effectiveness of the multigrid solver near the boundary is much less than for interior points. As a result, the asymptotic convergence rate of the entire solution grinds to a near halt.

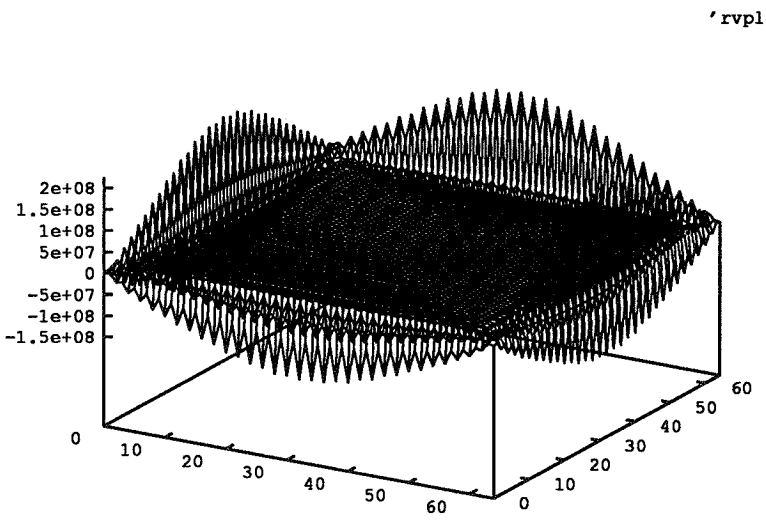


Figure 1: Boundary slowdown. A residual map (for $\Delta v = f$) after a number of multigrid cycles without special boundary treatment. The boundary residuals dominate.

A number of methods have been suggested to treat the slowness caused by the boundaries. Brandt in [1] has suggested adding extra relaxation sweeps at and around the boundaries, and has proved [2] that by doing so the efficiency dictated by the interior smoothing rates (as predicted by local mode analysis) can be attained. The additional work required is negligible relative to a full sweep on the entire domain. This idea has been partially implemented by Michel [9], who derived the adjusted residual transfers described above, and used them to measure convergence rates for lexicographic Gauss-Seidel relaxation. It is implemented here for Red-Black ordering. A different idea was suggested by Linden [8] and also by Papamanolis [10], who propose simultaneous relaxation of the boundary (v only, as u is given there) along with one neighboring interior point. Experimentally, this method has produced good results for grids up to 256 by 256. It only works, however, for the slower lexicographic relaxation ordering. After presenting these two methods, a new method fusing the two approaches will be presented, based on using simultaneous relaxation on a wider and deeper boundary strip. This method achieves the desired convergence rate for Red-Black ordering as well (but only for W cycles). Its main advantage (relative to the first method) lies in greater efficiency for massively parallel implementations.

Boundary Relaxations

In this method, a small number of relaxation sweeps are performed over the boundary and a small layer of points adjacent to the boundary. The extra work per relaxation sweep is $O(\sqrt{N})$ (N the total number of points), thus negligible relative to a full sweep ($O(N)$). The results to be presented were obtained using the following steps for a relaxation sweep:

- Perform the following $NREP$ times:
  1. Relax v on the boundary.
  2. Relax u and v on $DEPTH$ interior layers, starting nearest to the boundary.
- Using Red-Black ordering, relax the entire domain (including the boundary).

In all experiments, the domain is the unit square. Simultaneous relaxation is performed at interior points, meaning that new values are computed for both v and u (satisfying both equations of (3) there) each time the point is relaxed. $NREP$ and $DEPTH$ are parameters of the solver. The boundary-layer relaxation uses sequential ordering, although Red-Black ordering would probably serve just as well. Sensitivity of the algorithm to the order of execution (boundary relaxations before body, boundary relaxations from the border inward, etc.) was not rigorously tested. However, a casual sampling indicates that using Red-Black ordering for the boundary relaxations doesn't affect the results, while relaxing the boundary layer from the interiormost part to the boundary worsens performance somewhat.

Finest Grid	$NREP$	$DPTH$	V(1,1)	V(2,1)	W(2,1)	W(1,1)
64	0	0	< 1.5	< 2	< 4	< 4
64	1	2	3	5	10	4
	2	1	5	9	16	8
		2	5	10	16	9
		3	6	10	18	10
	3	1	2	3	8	6
		2	9	13	20	9
		3	10	13	18	13
128	2	2	5	9	16	9
	3	2	8	12	20	9
	3	3	8	13	18	13
256	2	2	5	8	16	9
	3	2	8	12	20	9
	3	3	8	11	18	13

Table 2: Cycling convergence rates.

Most of the experimentation was performed on a 64×64 finest grid, with other grids being tested to examine the effect of the gridsize on the necessary boundary

treatment. In all computations the coarsest grid was 4×4, except for the 256×256 finest grid, for which the coarsest grid was 8×8, as the software was designed for a maximum level depth of six. Table 2 sums it up.

Let us now try to make some sense out of the jumble of numbers in Table 2. First, it is evident for all types of cycle that a boundary layer of depth one (with any amount of boundary passes), or a single boundary pass (with boundary layer up to three deep) will not do (some of these results are not displayed in the table). For $(2,1)$ cycles (W and V), two passes with a width two layer bring convergence to about 80% of the periodic b.c. rate. Adding another pass improves results to 100%. For $V(1,1)$ cycles, nearly the same is true.

The situation is a little different for $W(1,1)$ cycles. Here, the maximal convergence rate obtained is about 13, slightly lower than the periodic b.c. rate of 15. Three passes are necessary over a boundary layer of width three.

For W cycles the results seem to be independent (or, at worst, imperceptibly dependent) on the gridsize. For V cycles, however, results deteriorate slowly with growing grids, requiring slowly increasing values for $NREP$ and $DEPTH$ (although the additional work remains negligible).

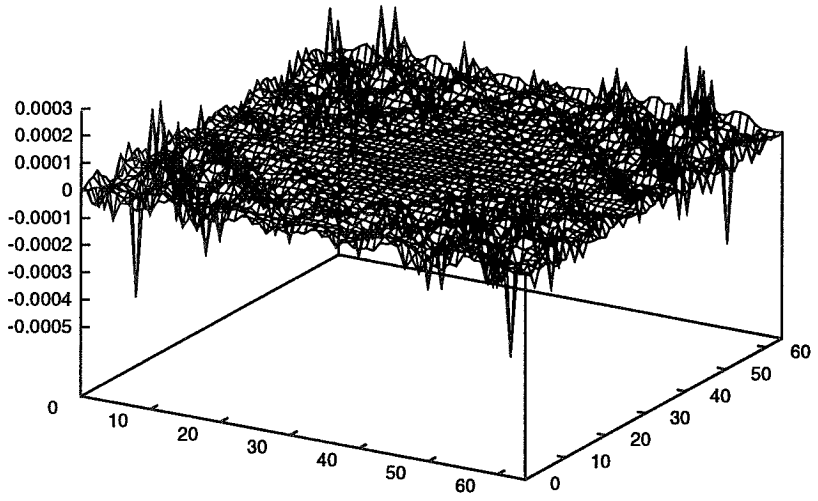


Figure 2: Treated boundary. A residual map (for $\Delta v = f$) after 30 multigrid cycles, adding boundary relaxations (three sweeps of a three-deep boundary layer). The boundary residuals are of the same magnitude as the interior residuals.

The added relaxations on the boundary alter the residual map shown in Figure 1. Figure 2 shows the result — with three boundary sweeps over a boundary layer three units wide, after thirty $W(2,1)$ cycles (long after convergence to machine accuracy is

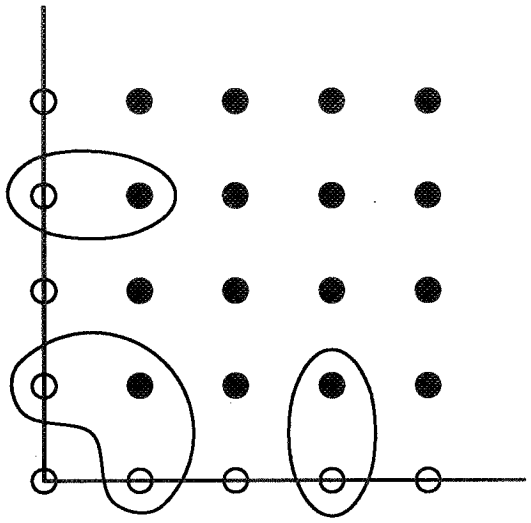


Figure 3: Joint relaxation (Linden's method).

attained, if desired; for the purpose of measuring asymptotic convergence rates, the error is artificially magnified between cycles), the residuals on the boundary are still of the same order of magnitude as those in the interior. Thus, the goal of converging as fast as a solver for the Poisson equation has been accomplished for $W(2,1)$, $V(2,1)$, and $V(1,1)$ cycles (and nearly so for $W(1,1)$ cycles).

It is interesting to note that the boundary treatment can be overdone (too much of a good thing...). After a point, adding more sweeps for a given layer-width diminishes performance, probably due to the jump in residual magnitude at the interface between the boundary strip and the interior.

This is the only method of the three presented here that reaches optimal performance for V as well as W cycles. The main drawback of this method is its unsuitability to massively parallel architectures, as body relaxation cannot proceed until the requisite number of boundary relaxations has been performed (rather than wait for two or three extra relaxations per sweep, it would be preferable to perform three complete cycles with no boundary treatment, which will, for $W(1,1)$ cycles, reduce the residuals by a factor of over thirty).

Joint Relaxation

An idea proposed independently by Linden [8] and Papamanolis [10] suggests relaxing each boundary v point together with its neighboring point (u and v), that is, solving a system for three simultaneous variables. At the corners, both near-corner v points are relaxed with the interior-corner point, a four-variable system. The variables joined in relaxation are shown in Figure 3.

The method works well only for W cycles, and only when the boundary layer is relaxed in lexicographic ordering. Table 3 shows convergence rates for an implementation of Linden's version of the algorithm, using both lexicographic and Red-Black

ordering on the boundary and on the interior. Poisson-solver convergence rates are obtained for lexicographic ordering, but not for Red-Black. The same is true for other W cycles, though the results are not displayed. A curious feature of the solver (using

Finest Grid	Lex.	RB
32	12	12
64	8-10	12
128	8-10	12
256	8-10	11

Table 3: Cycling convergence rates, $W(2,1)$ cycle, Linden's method, lexicographic and Red-Black ordering.

lexicographic ordering) is that it appears to have two stable rates of convergence, converging for a while at about 10 per cycle, then at about 8. Theoretically, the eight rate should dominate, as this is the smoothing rate for lexicographic ordering (in a $(2,1)$ cycle) predicted by local mode analysis. But this didn't happen for more than 150 cycles using a 128×128 finest grid (on a 200 cycle trial, the first 25 cycles converged at a rate of about 10 per cycle, the next 60 at about 8, the next 60 at about 11, and from then on at about 8).

Merging Methods

Using a method that, in a sense, merges the two ideas above, a new form of boundary treatment is obtained, more suitable for parallel implementations. Relaxation is performed in a manner similar to Linden's algorithm, although Red-Black ordering is used. A number of combinations were tested, one of which worked well. In the method which worked (Figure 4), three interior points were relaxed along with two boundary points (an eight variable system). At the corners the four cornermost interior points and their four boundary point neighbors combine to form a twelve variable system. What happens, in effect, is that nearly every point on the boundary layer is relaxed twice (simultaneously with the interior), once with the red interior points, and again with the black ones. The exceptions are two points in each corner which don't get a second relaxation.

Results are summarized in Table 4. In order to simulate parallel implementation of the method, the boundary solver does not use new values of u and v computed in the present half-sweep. Rather, the boundary relaxation during the 'red' part of the sweep uses values computed prior to the sweep, and during the 'black' part — values computed by the 'red' half.

Finally, it is worth noting that, with a bit of preprocessing, all the variables linked in joint relaxation can be relaxed in parallel, with the number of operations per variable proportional to the number of 'neighbors' of the system of equations. 'Neighbors'

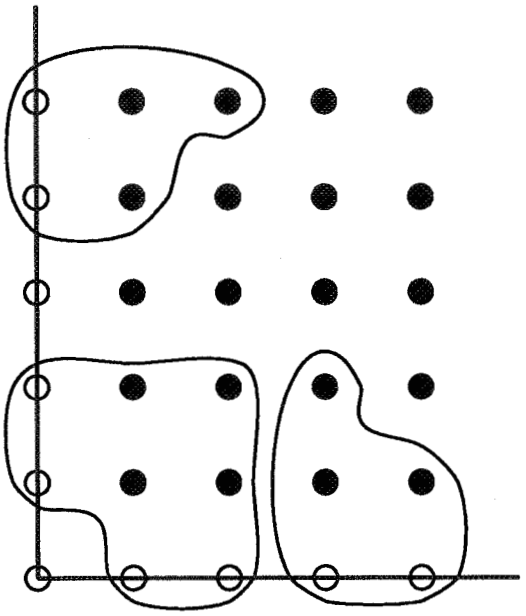


Figure 4: Joint relaxation. A single (v) variable at each boundary point and two variables at each interior point.

include the right hand sides of the equations[3], and all u and v variables participating in the equations but not solved for. Thus, the boundary system requires eighteen operations per variable (each 'operation' consisting of a constant multiplication and an addition), and the corner system twenty. Super parallel implementations can perform these computations in logarithmic time, as they are totally independent of each other. For comparison, relaxation of an interior point (after some economization) requires nine or ten operations to update u (the new value of v is computed in the first four). Here, each operation consists of a constant multiplication *or* an addition.

Finest Grid	$W(1,1)$	$W(2,1)$	$V(1,1)$	$V(2,1)$
32	14	20	6	11
64	14	20	5	9
128	14	20	4	7
256	14	20	3.5	6

Table 4: Cycling convergence rates, Red-Black ordering, boundary relaxed as in Figure 4.

[3]On the finest grid, where the right hand side doesn't change from cycle to cycle, a bit of preprocessing and some memory can reduce all the right hand side neighbors to a single operation per variable. The first and second methods will then each require only eleven operations per variable, and the corner system a mere nine.

FMG Convergence

Using FMG and the boundary treatments outlined above, the biharmonic equation can be solved in a single cycle to a point where the dominant term in the error (for some norms) is due to the truncation error (the error due to the approximation of the equation on a grid) rather than algebraic error (the error in the solution of the equation). Table 5 shows the error in the solution obtained after one, two, and three multigrid cycles using the FMG algorithm to provide the initial guess for the finest level. Cubic interpolation is used to transfer the coarse grid solution (u and v) to the fine grid. The differential solution for the system tested was

Grid	Norm	$V(1,1)$ Cycle	One Cycle	Two Cycles	Three Cycles	$W(2,1)$ Cycle
64	L_∞	3.6e-6	5.6e-6	7.5e-6	7.3e-6	6.2e-6
	H_1	1.1e-7	1.9e-7	2.7e-7	2.6e-7	2.1e-7
	H_2	5.0e-8	4.8e-8	3.1e-8	2.9e-8	3.5e-8
128	L_∞	4.0e-7	1.4e-6	1.9e-6	1.8e-6	1.6e-6
	H_1	9.0e-9	2.4e-8	3.3e-8	3.2e-8	2.7e-8
	H_2	6.6e-9	4.5e-9	2.1e-9	1.9e-9	2.8e-9
256	L_∞	1.3e-7	3.5e-7	4.7e-7	4.6e-7	3.9e-7
	H_1	1.0e-9	3.0e-9	4.2e-9	4.1e-9	3.4e-9
	H_2	4.7e-10	4.0e-10	1.9e-10	1.8e-10	2.7e-10

Table 5: FMG. Error (relative to the differential solution) after one $V(1,1)$, one, two and three $W(1,1)$ cycles, and one $W(2,1)$ cycle.

$x^2(1-x)^2y^2(1-y)^2$, and the errors in the solutions on each grid are measured relative to this function. Three error norms (or seminorms) are shown: L_∞, H_1 ($\sqrt{\int u_x^2 + u_y^2}$), and H_2 ($\sqrt{\int u_{xx}^2 + u_{yy}^2 + 2u_{xy}^2}$).

Clearly, after even one $V(1,1)$ cycle[4], the error (L_∞ and H_1) is primarily due to truncation - in fact, in this particular case the algebraic error happens to cancel part of the truncation error, as solving the system to a higher degree of accuracy increases the distance from the differential solution. This is not true for the H_2 error, which does indeed get significantly reduced by further cycles. Using a higher order of interpolation to transfer the initial guess to the fine grid should correct this.

The number of operations (per fine grid point) required to solve the biharmonic equation will therefore equal the work necessary to perform a single $V(1,1)$ cycle on each grid (coarsest to finest). Assuming 9 operations (constant multiply or add) per point for each relaxation sweep (10 on coarser grids), about 16 for coarsening and interpolation combined (note that residuals need be transferred from Red relaxation points only), and neglecting the work added by the boundary treatment, this gives a total of less than 70 operations per point to solve.

[4]These results were obtained using joint boundary relaxation. Using extra boundary sweeps gives a similar but slightly better solution. However, for larger grids, it may be necessary to use boundary sweeps, use more points in joint relaxation, or use $W(1,1)$ cycles.

REFERENCES

[1] A. Brandt, "Guide to Multigrid Development, " in *Multigrid Methods* (W. Hackbush and U. Trotenberg ed.), Springer-Verlag, 1982, 210-312.

[2] A. Brandt, "Rigorous Quantitative Analysis of Multigrid," *SIAM J. Num. Anal* **31**, 1994, 1695-1730.

[3] S. C. Brenner, "An Optimal-Order Nonconforming Multigrid Method for the Biharmonic Equation," *SIAM J. Numer. Anal.* **26**, 1989, 1124-1138.

[4] P. G. Ciarlet and P. A. Raviart, "A Mixed Method for the Biharmonic Equation," in *Mathematical Aspects of Finite Elements in Partial Differential Equations* (C. de Boor ed.), Academic Press, 1974 , 125-145.

[5] J. Dym, "Multilevel Methods for Early Vision," Ph. D. Thesis, Weizmann Institute of Science, 1994.

[6] R. Glowinski, "Approximations externes, par éléments finis de Lagrange d'ordre un et deux, du problème de Dirichlet pour l'opérateur biharmonique. Methods itératives de résolution des problèmes approchés," in *Topics in Numerical Analysis* (J. J. H. Miller ed.), Academic Press, 1973, 123-171.

[7] M. R. Hanisch, "Multigrid Preconditioning for the Biharmonic Dirichlet Problem," *SIAM J. Numer. Anal.* **30**, 1993, 184-214.

[8] J. Linden, "A Multigrid Method for Solving the Biharmonic Equation on Rectangular Domains," Arbeitspapiere der GMD 143, 1983.

[9] R. Michel, "Boundary Techniques for the Numerical Solution of Elliptic Systems of P.D.E's," M. Sc. Thesis, Weizmann Institute of Science, 1985.

[10] G. Papamanolis, 'Multigrid Methods in Fluid Dynamics," M. Sc. Thesis, University of Wales, Aberystwyth, 1984.

Page intentionally left blank

MULTIGRID ACCELERATION OF TIME-ACCURATE DNS OF COMPRESSIBLE TURBULENT FLOW*

Jan Broeze, Bernard Geurts, Hans Kuerten, and Martin Streng
University of Twente, Dept. of Applied Mathematics,
P.O. Box 217, 7500 AE Enschede, The Netherlands

ABSTRACT

An efficient scheme for the direct numerical simulation of 3D transitional and developed turbulent flow is presented. Explicit and implicit time integration schemes for the compressible Navier-Stokes equations are compared. The nonlinear system resulting from the implicit time discretization is solved with an iterative method and accelerated by the application of a multigrid technique. Since we use central spatial discretizations and no artificial dissipation is added to the equations, the smoothing method is less effective than in the more traditional use of multigrid in steady-state calculations. Therefore, a special prolongation method is needed in order to obtain an effective multigrid method.

This simulation scheme was studied in detail for compressible flow over a flat plate. In the laminar regime and in the first stages of turbulent flow the implicit method provides a speed-up of a factor 2 relative to the explicit method on a relatively coarse grid. At increased resolution this speed-up is enhanced correspondingly.

INTRODUCTION

Multigrid methods have proven to be very successful when computing steady solutions to the Reynolds-averaged Navier-Stokes equations [6,12]. In these equations a turbulence model is introduced and an approximation for the mean turbulent flow field is obtained. Many turbulent flows are only statistically stationary, however, and the actual solution is strongly time dependent. The development of numerical simulation methods for the time accurate simulation of turbulent flow forms a subject of intensive research (see [3,4,9,10,13]). In particular the transition from laminar to turbulent flow and the early stages of fully developed turbulence in relatively simple geometries are presently accessible to time-accurate numerical simulation.

*This work was supported by the J.M. Burgers Centre and by the Netherlands Organization for Scientific Research (NWO).

DNS forms a key tool for computing detailed and reliable results for turbulent flow in simple geometries, which can subsequently be used in the validation of numerical methods and sub-grid models for large eddy simulations. In this paper we focus on an efficient higher order accurate method for direct numerical simulation (DNS) of compressible flow. In this paper results will be illustrated for the compressible flow over a flat plate.

Because of the large variety of length scales present in high-Reynolds turbulent flows, a large number of grid points is required. The grid should be chosen such that the relevant modes with smallest length scales can still be adequately represented, resulting in very fine meshes. The time step is limited by accuracy requirements and stability conditions unless an absolutely stable time integration method is applied. In general, the stability conditions are far more restrictive than the accuracy requirements, especially in the laminar regime. Stability conditions for explicit time integration methods lead to a linear relation between the grid size and the time step if the convective terms in the equations are most restrictive. Thus, the required number of time steps is proportional to n (the number of grid points in each grid direction). In principle, absolutely stable (thus implicit) time integration methods are more suitable for this type of problem, since no stability restrictions are imposed on the time step. However, implicit methods are more expensive per time step. Hence, effective techniques are required for a fast solution of the nonlinear system of equations resulting from the implicit time discretizations in order to render these methods useful.

Summarizing, the following dilemma is observed. Application of explicit time integration methods leads to a large number of (relatively cheap) time steps, with a total number of operations $a\,n^4$. With an implicit scheme a relatively small number of (expensive) implicit time steps is required, leading to $b\,n^3$ operations. However, in general b is considerably larger than a.

The main purpose of our study is the development of efficient tools for solving the system of equations that arises from application of an implicit time integration scheme to the compressible Navier-Stokes equations. The method presented in this paper is based on the work by Jameson [6,7] and Melson *et al.* [12]. For Reynolds-averaged Navier-Stokes (RaNS) equations they have proposed an iterative-implicit method which is based on a multigrid technique, leading to a considerable speed-up in comparison with explicit methods. The RaNS equations contain a lot of dissipation, which leads to a fast convergence of the relaxation method. In our equations, however, the dissipation is very small. As a result, we obtain a smaller speed-up than Melson *et al.*

This paper is set up as follows. In the following section, a general description of the equations is given. Numerical solution techniques for the problem (including a multigrid technique) are presented in section 3. In section 4 computational results will be presented. Finally, we will give some conclusions.

GOVERNING EQUATIONS

The equations describing the flow are the well-known Navier-Stokes equations, which represent conservation of mass, momentum and energy. In terms of dimensionless variables (density ρ, velocity components u_j and energy density e) these equations have the form (the summation convention is used):

$$\partial_t \rho + \partial_j(\rho u_j) = 0 \tag{1}$$

$$\partial_t(\rho u_k) + \partial_j(\rho u_k u_j) + \partial_k p - \partial_j \sigma_{kj} = 0 \qquad k = 1,2,3 \tag{2}$$

$$\partial_t e + \partial_j((e+p)u_j) - \partial_j(\sigma_{ij}u_i - q_j) = 0 \tag{3}$$

Here ∂_t and ∂_j denote partial differentiation with respect to time and the coordinate x_j, respectively. The pressure p is given by

$$p = (\gamma - 1)(e - \frac{1}{2}\rho u_i u_i) \tag{4}$$

in which γ denotes the adiabatic gas constant, which is set to $\gamma = 1.4$. The viscous stress tensor σ_{ij} is a function of the velocity components u_j:

$$\sigma_{ij} = \frac{1}{Re}(\partial_j u_i + \partial_i u_j - \frac{2}{3}\delta_{ij}\partial_k u_k) \tag{5}$$

where Re is the Reynolds number (the fluid viscosity is taken constant). Furthermore, q_j represents the viscous heat flux, given by

$$q_j = -\frac{1}{(\gamma - 1)RePrM_r^2}\,\partial_j T \tag{6}$$

where Pr is the Prandtl number, for which we use $Pr = 0.72$, and M_r is the reference Mach number. The temperature T is given by the ideal gas law:

$$T = \gamma M_r^2 \frac{p}{\rho} \tag{7}$$

In the Navier-Stokes equations (1-3), two types of fluxes can be distinguished. The convective fluxes consist of the first order spatial derivatives in the Navier-Stokes equations. These are of hyperbolic type and in Von Neumann analysis of the linearized equations, they give rise to imaginary eigenvalues. The viscous fluxes are parabolic and add dissipation to the system. This dissipation is $\mathcal{O}(1/Re)$.

The behavior of the solution of the Navier-Stokes equations roughly can be characterized as follows. The nonlinear terms in the convective fluxes provide a continuous generation of modes with a small length scale from the components with a larger length scale. On the other hand, the dissipative fluxes add a certain damping to the system. This damping is very small for the components with a large length-scale, but it is larger for the small-length-scale components. In the transitional stage from laminar to turbulent flow, small disturbances in a laminar flow give rise to growth of

large-scale eddies (which correspond to the most unstable modes in linear stability theory). These eddies generate eddies with smaller length scales. This continuous flow of energy to the eddies with smaller length scales is truncated at the scale where the dissipation counterbalances the growth effects, so that a statistically stationary turbulent flow is obtained. This "viscous length scale" strongly depends on the Reynolds number. In the turbulent regime a broad spectrum of different modes in the flow develops.

NUMERICAL METHOD

In this section the discretization of the spatial derivatives and the explicit and iterative-implicit time integration methods for our problem will be discussed.

Spatial discretization

For the spatial derivatives in the equations, fourth order accurate central five-point difference schemes are used. Since artificial dissipation may seriously influence the solution during the transition from laminar to turbulent flow, the schemes are devised in such a way that no artificial dissipation is required. Odd-even decoupling is prevented by using a filtering procedure that just eliminates the shortest modes, see e.g. [4].

Explicit time integration

After discretizing the spatial derivatives in the governing equations, the equations take the following form (with discrete state vector U):

$$\partial_t U + f(U) \;=\; 0 \tag{8}$$

In the numerical solution of this problem, we denote the numerical solution at time level t_n by U^n.

We have implemented a second order compact-storage four-stage Runge-Kutta method. The method is suitable for our problem, since the stability region contains a considerable part of the imaginary axis (up to $2\sqrt{2}\,i$). Thus, this method gives stable results if the size of the time step satisfies the CFL condition:

$$\Delta t\, \lambda_m \;\leq\; C_{CFL} \tag{9}$$

with $C_{CFL} \approx 2\sqrt{2}$. The largest eigenvalue λ_m of the discrete linearized convective flux is given by

$$\lambda_m \;=\; A\left(\frac{|u_1|}{\Delta x_1} + \frac{|u_2|}{\Delta x_2} + \frac{|u_3|}{\Delta x_3} + \sqrt{\frac{\gamma p}{\rho}}\sqrt{\frac{1}{\Delta x_1^2} + \frac{1}{\Delta x_2^2} + \frac{1}{\Delta x_3^2}}\right) \tag{10}$$

with $A = \frac{2}{3}\sqrt{-6+4\sqrt{6}} + \frac{1}{6}\sqrt{-39+16\sqrt{6}} \approx 1.37$ if fourth order accurate central five-point finite difference approximations are used on an orthogonal equidistant grid. Thus, increasing the number of grid points leads to a proportional reduction of the time step.

Iterative-implicit time integration methods

In order to speed up the solution method, an implicit time integration scheme has been applied. A-stable methods (i.e., those that have a region of stability which includes the whole of the left half-plane (also referred to as absolute stability)) are preferred, so that the time step is not restricted by stability conditions, but only by physics. An iterative procedure is applied for the solution of the system of equations resulting from the implicit scheme, thus the approach is called an iterative-implicit method.

We will only consider implicit linear multistep schemes. Because of the complexity of the equations and the large number of points involved in our problems, more advanced schemes are not considered here. The order of A-stable linear multistep methods cannot exceed two (see [2,11]). Suitable methods are Backward Euler, the Trapezoidal Rule, and the two step Backward Differentiation Formula, BDF(2). Since Backward Euler is only first order accurate and generates considerable numerical damping, we have decided not to use it. The Trapezoidal Rule is the second-order A-stable linear multistep method with smallest error constant [2,11]. However, since periodic eigenfunctions are not damped, extra smoothing is required in many applications. The BDF(2) method is preferred, because it is less sensitive and has a larger stability region.

For eq.(8), BDF(2) with constant Δt is defined by

$$3\,U^{n+1} - 4\,U^n + U^{n-1} = -2\,f(U^{n+1})\,\Delta t \tag{11}$$

In order to solve eq.(11), it is written as

$$a_0 V + f(V) = g \tag{12}$$

where V stands for the unknown solution U^{n+1}, $g = (4U^n - U^{n-1})/(2\Delta t)$ is a known forcing function and $a_0 = 3/(2\Delta t)$ for this time integration method. Other implicit time integration schemes can be cast in the same form (12) by adjusting the constants and forcing functions.

An iterative method for solving eq.(12) consists of the following steps:

1. computation of a starting solution V^0,
2. relaxation method for improving the solution,
3. truncation of the relaxation method if the desired accuracy is achieved.

Our approach for these steps will be presented below.

Starting solution

It is clear from the above that the solutions at two previous time levels are required to calculate the solution at a new time level. This is inherent to the second order accurate discretization of the time derivative in eq.(11). The availability of solutions at previous time levels can also be exploited to obtain a good starting solution at the new time level. The better the starting solution corresponds with the solution to eq.(11), the smaller the amount of work that is necessary to calculate the solution within the required accuracy. A suitable starting solution is obtained from extrapolation of the solution from previous time levels. For constant time steps Δt quadratic extrapolation yields

$$V^0 \;=\; 3\,U^n \;-\; 3\,U^{n-1} \;+\; U^{n-2} \tag{13}$$

Another second order extrapolation method uses the time derivative of U given by the function f:

$$V^0 \;=\; U^{n-1} \;-\; 2\,f(U^n)\,\Delta t \tag{14}$$

A second order extrapolation method with a number of similar terms in the truncation error as in the truncation error of the BDF(2) formula is

$$V^0 \;=\; \frac{1}{3}\left(4\,U^n - U^{n-1} - 4\,f(U^n)\,\Delta t + 2\,f(U^{n-1})\,\Delta t\right) \tag{15}$$

The truncation errors of eq.(14) and BDF(2) are very different. As a result, more relaxations are required if extrapolation (14) is used than if eq.(13) or eq.(15) is applied. The choice of either eq.(13) or eq.(15) does not have a large influence on the required number of relaxations.

Iteration method; application of multigrid

A standard method to solve equations of the form (12) is the Newton-Raphson iteration method. In this method, a linearization of the flux vector around the known state U^n is used, see e.g. [13]. However, application of this method goes at the expense of a large matrix inversion.

Multigrid methods [1,5] are often applied for the efficient computation of steady-state solutions to RaNS equations. Because of the large number of grid points and the large variety of typical length scales in the solution, application of these methods leads to significant accelerations compared to classical iteration methods if suitable smoothing methods are used. In fact, our problem (12) is of the same form; hence we can utilize the same technique each time step. This is described below.

Transfer operators

The solution is restricted to coarser grids by injection, and the defect vector by full weighting.

A special treatment of the prolongation is required. Basically, the correction is prolonged to the finer grid by means of trilinear interpolation. This prolongation operator works well for stationary flow simulations, since the fine grid operator indeed satisfies the requirement of damping the high frequency components in the error. In the present solver, however, the high frequency components which may be created by the prolongation process are very slowly damped since the discretization method does not contain artificial damping. Therefore, after every prolongation first the shortest modes are removed from the correction by applying a filtering operator to the corrections. This filter eliminates the shortest modes.

Smoothing method

The rapidly varying eigenfunctions (so-called rough eigenfunction, see [14]) cannot be represented well on coarse grids. Therefore, an effective smoothing method is required.

A common technique for the computation of steady-state solutions to the Navier-Stokes equations is solving the time dependent equations with multistage methods (see e.g. [7]). We have chosen a similar approach for our problem (12). In order to solve (12), we find the steady state solution of the following pseudo time evolution equation:

$$\partial_\tau V + a_0 V + f(V) = g \tag{16}$$

The advantage of writing the problem in this form is that it has the good stability properties of the implicit time integration method, whereas the flexibility of explicit time integration schemes is maintained. Furthermore, convergence acceleration methods can be applied in a manner similar to steady-state calculations.

We have chosen the following second order accurate Runge-Kutta method:

$$\begin{aligned}
V_0 &= V^m \\
(1+\tfrac{1}{4}a_0\Delta\tau)V_1 &= V_0 - \tfrac{1}{4}(f_c(V_0)+f_d(V_0)-g)\Delta\tau \\
(1+\tfrac{1}{6}a_0\Delta\tau)V_2 &= V_0 - \tfrac{1}{6}(f_c(V_1)+f_d(V_0)-g)\Delta\tau \\
(1+\tfrac{3}{8}a_0\Delta\tau)V_3 &= V_0 - \tfrac{3}{8}(f_c(V_2)+f_d(V_2)-g)\Delta\tau \\
(1+\tfrac{1}{2}a_0\Delta\tau)V_4 &= V_0 - \tfrac{1}{2}(f_c(V_3)+f_d(V_2)-g)\Delta\tau \\
(1+a_0\Delta\tau)V_5 &= V_0 - (f_c(V_4)+f_d(V_4)-g)\Delta\tau \\
V^{m+1} &= V_5
\end{aligned} \tag{17}$$

in which $f = f_c + f_d$. The dissipative part (f_d) of the flux vector is only evaluated at a few stages, as proposed by e.g. Jameson [7, 8]. This both saves calculation time and increases the stability.

Furthermore, in this time stepping scheme, the linear term $a_0 V$ in eq.(16) is treated implicitly. This is easily possible, as the term is diagonal, and useful, since it improves the stability of the pseudo time stepping method: since $a_0 > 0$ the stability function is modified, which leads to a larger stability region and a considerable reduction of the amplification factor, see e.g. [12].

If $a_0 = 0$, the CFL number can be taken to be 4.0. From a linearization of the stability function around $\Delta\tau\ \lambda_m = 4.0$, the following conditions can be derived for small $a_0/\lambda_m > 0$:

$$\begin{array}{lcl} \Delta\tau\ (\lambda_m - a_0) & < & 4.0 \\ \Delta\tau\ (\lambda_m + \frac{1}{12} a_0) & < & 4.5 \end{array} \tag{18}$$

Other convergence acceleration techniques

The convergence is more accelerated by the application of local pseudo time stepping. Since a steady state equation is solved, the pseudo time step $\Delta\tau$ need not to be equal in each point. The maximum allowed $\Delta\tau$ is chosen in each point from eq.(18).

At this moment, we are testing a Newton-Raphson type approach. This approach is based on a linearization of the function f around the solution V^m (see eq.(16)). Using upwind discretizations for modifications of the solution, then, the characteristic variables associated with the largest eigenvalue are solved implicitly. First test results indicate that a speed-up of 10 to 20% can be obtained.

Truncation of the relaxation process

For the time being we have chosen the following approach. The iterations are truncated if the residual is below a prescribed value. The maximum value is chosen so that the truncation error is smaller than the discretization error of the time integration method.

APPLICATION TO A TRANSITIONAL FLOW

In this section we present some results from application of the techniques described above to a transitional wall-bounded flow.

The flow is computed in a rectangular domain, with no-slip isothermal wall conditions at the wall, symmetry conditions at the upper boundary, and periodicity in the horizontal directions. The initial solution consists of the similarity solution for a compressible boundary layer combined with a small-amplitude disturbance, consisting of a number of unstable modes which are obtained from linear stability theory.

The computations presented here were done with the iterative-implicit time integration method. In the multigrid process, V-cycles are used with 1 pre-relaxation

and 2 post-relaxations. The relaxation process is truncated if the residual has become less than a certain prescribed value. This value is chosen such that the resulting truncation error is smaller than the errors due to the discretizations.

The equations are discretized on a domain with $64 \times 64 \times 64$ grid points, which is adequate at the stage of transition and quite coarse in the turbulent regime.

Discretization errors

First we will show that in the laminar regime the small time steps required for stability of the explicit time integration method are not necessary for accuracy.

In the laminar regime, only disturbances with relatively large length-scales are present. The following table shows the relative errors due to the spatial discretization for the growth rate of the most unstable mode for different grid densities. ε_1 and ε_2 are the relative errors of the growth rate at two different locations: in the boundary layer and further in the domain, respectively.

n	ε_1	ε_2
32	0.05	0.005
64	0.005	0.0003
128	0.0003	0.0000

TABLE 1
Relative errors in growth rate of most unstable mode
due to the spatial discretization, at two locations in the flow
(n stands for the number of grid points in each grid direction)

The results in Table 1 show the 4th order accuracy of the spatial discretization method.

Discretization errors due to the time integration are given in Table 2.

Δt	ε_1	ε_2
0.05	0.0000	0.0000
0.10	0.0002	0.0000
0.25	0.001	0.0003
0.50	0.003	0.001
1.00	0.01	0.005

TABLE 2
Relative errors in growth rate of most unstable mode
due to the time discretization, at two locations in the flow

It is clear from these data that for a $64 \times 64 \times 64$-grid, a time step $\Delta t = 0.50$ may be chosen which leads to an error in the growth rate due to the time integration

comparable to the error arising from the spatial discretization. This value is in large contrast with the size of the time step limit for the explicit time integration method: for this grid $\Delta t \leq 0.04$ is required. The large discrepancy between these two values of the time step is the main motivation for this study. In a later stage, when modes with smaller length scales become more important, this discrepancy becomes smaller. This is sketched in Figure 1.

Since the number of operations per pseudo time step is proportional to the number of grid points, the amount of work done on the coarse grids can be neglected. Furthermore, the required amount of CPU time for one explicit time step is approximately equal to the time for one pseudo time step with eq.(17) on the finest level. Thus, the ratio of the CPU time per implicit and explicit time step is the measured number of pseudo time steps on the finest grid. Typical numbers will be given in the following subsection.

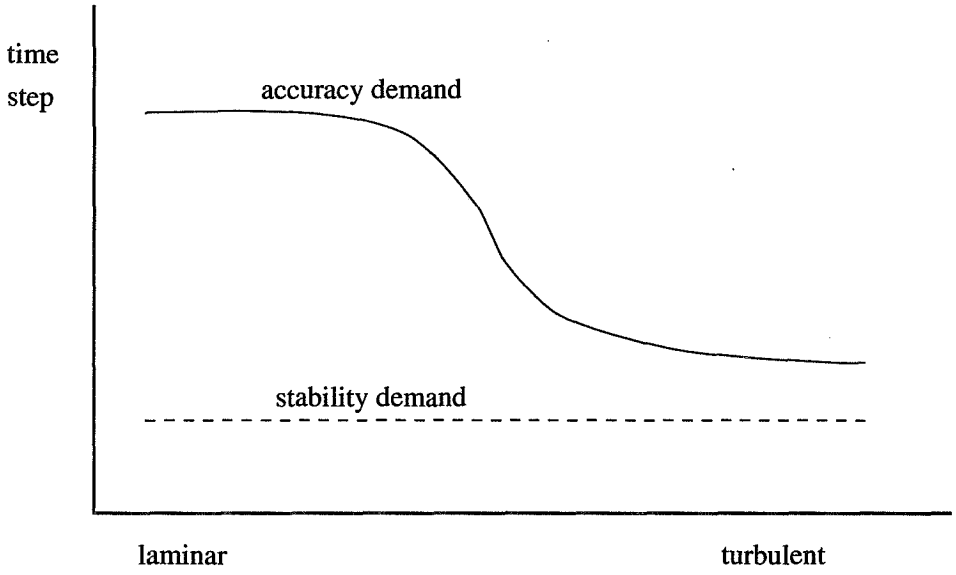


Figure 1: Typical behavior of the time step limit based on accuracy and stability requirements in the transition from laminar to turbulent flow

From a comparison of numerical results obtained with the explicit and the implicit scheme, we conclude that the differences are very small in the laminar regime. Figure 2 illustrates that with fixed time step $\Delta t = 0.5$ the results are accurate up to $t \approx 2250$. The accuracy of the implicit time integration method is increased if smaller values are used for the implicit time step. Thus, choosing time steps that are larger than the value prescribed by the stability condition for the explicit time integration method is allowed in the laminar regime.

Comparison of the efficiency of the explicit and implicit method

Various criteria can be used for determining the size of the time step. The application of the iterative-implicit time integration method is illustrated with the following

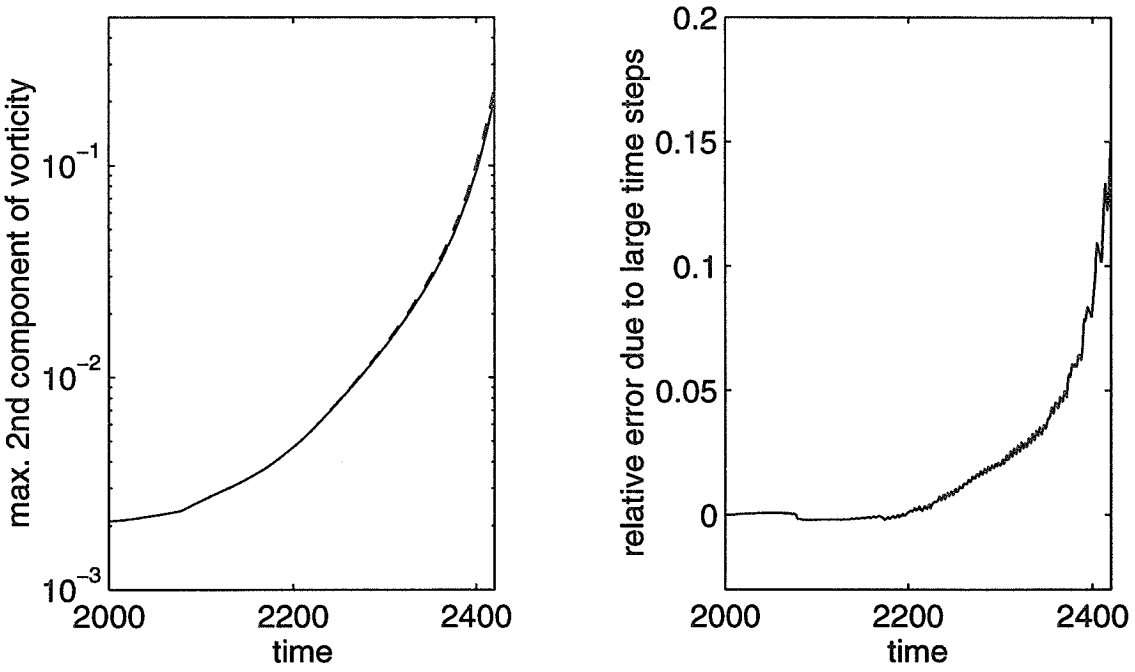


Figure 2: (a) Comparison of the development of the maximum value of the second vorticity component with explicit time integration (solid line) and with the implicit method ($\Delta t = 0.5$). (b) Relative error due to the large time step

choices: (a) choose a fixed time step ($\Delta t = 0.5$), or (b) vary the time step so that the system of equations can be solved in about 2 V-cycles. Figure 3 shows a comparison of the work involved with both choices and with the explicit time integration method.

The differences between the amount of work with fixed and variable time steps (both with the iterative-implicit time integration method) can be explained as follows. The iterative-implicit time integration method outlined above is similar to a predictor-corrector method. Predictor-corrector methods have a bounded stability region. The stability can be increased by choosing smaller time steps or by increasing the number of corrections. For a large number of predictor-corrector methods, the length of the part of the imaginary axis in the stability region increases less than linearly with the number of corrections, see e.g. the stability regions for Adams-Bashforth methods in [11]. Therefore, we expect that reducing the time step size Δt is cheaper than increasing the number of cycles.

The stability requirements for the explicit time integration method lead to a maximum time step $\Delta t \approx 0.04$, so that 250 explicit time steps are needed per 10 time units. Apparently, the implicit time integration scheme is cheaper than the explicit scheme if the implicit time step is sufficiently reduced. In our experiences for $t > 2400$ the grid should be reduced for a sufficient representation of the shortest modes; again our implicit scheme is more efficient than the explicit scheme.

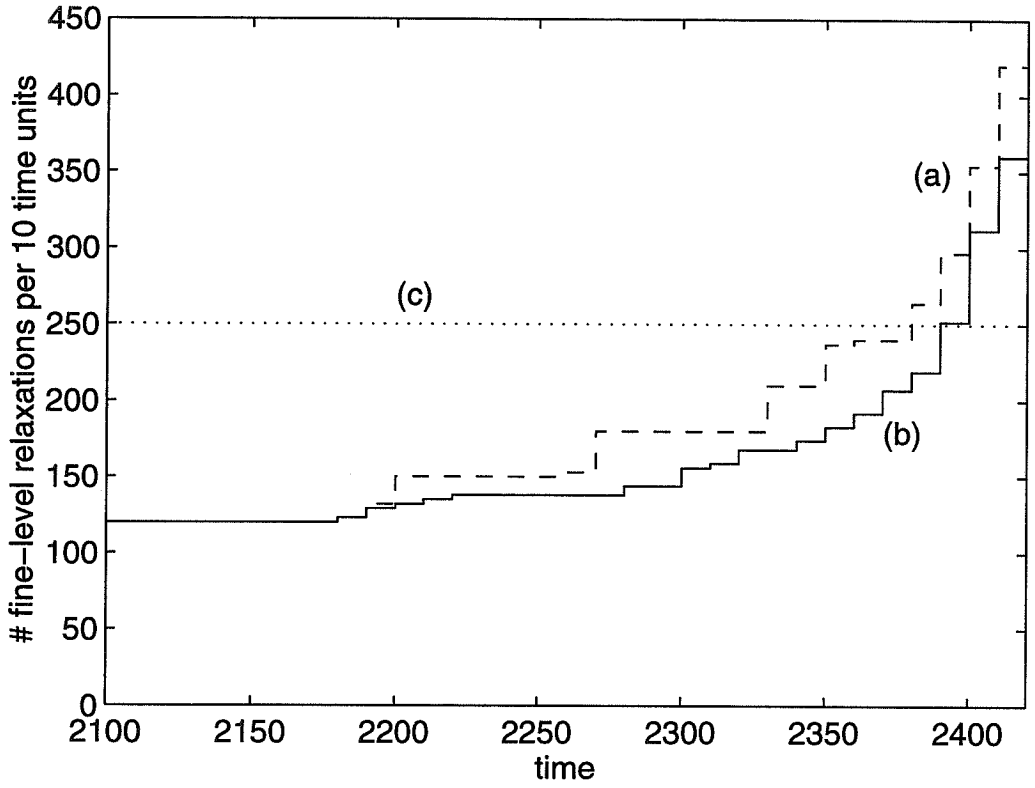


Figure 3: Number of pseudo time steps per 10 time units on the finest grid with (a) a fixed time step and (b) a variable time step compared with (c) the number of time steps for the explicit time integration method. In this time interval, the transition from laminar to turbulent flow occurs.

Finally, it is noted that if the multigrid method is not used, solving the system of equations takes about 5 times more CPU time. Thus, the use of this method is decisive for the success of the implicit scheme.

CONCLUSIONS

We have compared the application of an explicit and an iterative-implicit time integration scheme to time-accurate DNS of compressible turbulent flow. Convergence acceleration techniques such as multigrid are crucial for an effective iterative solution of the system of equations. For the application presented in this paper, the iterative-implicit method is faster than the explicit solver. However, due to the small amount of dissipation in the equations a smaller speed-up is obtained than in methods for the RaNS equations.

REFERENCES

[1] Brandt, A.: Multi-level adaptive solutions to boundary value problems, *Math. Comput.*, Vol. 31, pp. 333-390 (1977).

[2] Dahlquist, G.: A special stability problem for linear multistep methods, *BIT*, Vol. 3, pp. 27-43 (1963).

[3] G. Erlebacher and M.Y. Hussaini: Numerical experiments in supersonic boundary-layer stability, *Phys. Fluids A*, Vol. 2, No. 2, pp. 94-104 (1990).

[4] Geurts, B., Vreman, B. and Kuerten, H.: Comparison of DNS and LES of transitional and turbulent compressible flow: flat plate and mixing layer, In: *Application of direct and large eddy simulation to transition and turbulence, Proceedings of the 74th Fluid Dynamics Symposium* (Agard Conference Proceedings 551), Chania, Crete, Greece (1994), pp. 5.1-5.14.

[5] Hackbusch, W.: *Multi-grid methods and applications*, Springer-Verlag, Berlin (1985).

[6] Jameson, A.: Transonic flow calculations for aircraft, *In: Numerical methods in fluid dynamics*, F. Brezzi (ed.), (*Lecture Notes in Mathematics* 1127), Springer-Verlag, Berlin, pp. 156-242 (1985).

[7] Jameson, A.: Time dependent calculations using multigrid, with application to unsteady flows past airfoils and wings, *AIAA Paper* 91-1596 (1991).

[8] Jameson, A and Baker, T.J.: Multigrid solution of the Euler equations for aircraft configurations, *AIAA-paper* 84-0093 (1984).

[9] Kuerten, H. and Geurts, B.: Multigrid acceleration of block structured compressible flow solver, *J. Engg. Math.*, Vol. 29, pp. 11-31 (1995).

[10] L. Kleiser and T.A. Zang: Numerical simulation of transition in wall-bounded shear flows, *Ann. Rev. Fluid Mech.*, Vol. 23, pp. 495-537 (1991).

[11] Lambert, J.D.: *Numerical Methods for ordinary differential equations*, John Wiley & Sons, Chichester, (England), (1991).

[12] Melson, N.D., Sanetrik, M.D., and Atkins, H.L.: Time-accurate Navier-Stokes calculations with multigrid acceleration, *Sixth Copper Mountain Conference on Multigrid Methods* (NASA CP-3224), Part 2, pp. 423-437 (1993).

[13] Rai, M.M. and Moin, P.: Direct numerical simulation of transition and turbulence in a spatially evolving boundary layer, *J. Comp. Phys.*, Vol. 109, pp. 169-192 (1993).

[14] Wesseling, P.: *An introduction to multigrid methods*, John Wiley & Sons, Chichester, (England), (1992).

Page intentionally left blank

FIRST-ORDER SYSTEM LEAST SQUARES FOR VELOCITY-VORTICITY-PRESSURE FORM OF THE STOKES EQUATIONS, WITH APPLICATION TO LINEAR ELASTICITY

ZHIQIANG CAI*, THOMAS A. MANTEUFFEL†, AND STEPHEN F. McCORMICK‡

Abstract. In this paper, we study the least-squares method for the generalized Stokes equations (including linear elasticity) based on the velocity-vorticity-pressure formulation in $d = 2$ or 3 dimensions. The least-squares functional is defined in terms of the sum of the L^2- and H^{-1}-norms of the residual equations, which is similar to that in [6], but weighted appropriately by the Reynolds number. Our approach for establishing ellipticity of the functional does not use ADN theory, but is founded more on basic principles. We also analyze the case where the H^{-1}-norm in the functional is replaced by a discrete functional to make the computation feasible. We show that the resulting algebraic equations can be uniformly preconditioned by well-known techniques.

Key words. least squares, Stokes

AMS(MOS) subject classifications. 65F10, 65F30

1. Introduction. Recently, there has been substantial interest in the use of least-squares principles for numerical approximation of the incompressible Stokes and Navier-Stokes equations, especially those based on vorticity (more precisely, velocity-vorticity-pressure); for example, see [5, 12, 13, 14, 19]. Its attractions include accurate approximation to meaningful physical quantities, formulation of a well-posed minimization principle, elimination of the need for artificial stablization techniques, and freedom in the choice of finite element spaces (which are not subject to the LBB condition). The computational results provided in these papers indicate that such methods have great promise. However, they do not yield optimally accurate approximations for the case of Dirichlet boundary conditions (see the analysis in [6]). In recent work by Bochev and Gunzburger [6], the ADN approach (see [2]) was extended to the vorticity formulation of the Stokes equations with rigorous error analysis. The least-squares functional is defined to be the sum of squares of the norms of the residual of each equation, where the norms are determined by the indices assigned to each equation by the ADN theory (see [1]). To be specific, consider the two-dimensional stationary Stokes equations with Dirichlet boundary conditions. Then ADN theory was used in [6] to show that the least-squares functional $\|\mathbf{f}-(\nu\nabla^{\perp}\omega+\nabla p)\|_q^2+\|\nabla\times\mathbf{u}-\omega\|_{q+1}^2+\|\nabla\cdot\mathbf{u}\|_{q+1}^2$ is equivalent to the sum of squared norms of each variable, $\|\mathbf{u}\|_{q+2}^2+\|\omega\|_{q+1}^2+\|p\|_{q+1}^2$, for all $q\in R$ and $\mathbf{f}=\mathbf{0}$. In particular, they consider the above functional with $q=0$, then replace the H^1-norms by mesh-dependent L^2-norms, $h^{-2}\|\cdot\|_0^2$ (see also [2]). This mesh-dependent least-squares approach yields optimally accurate approximations for each variable with respect to approximation subspaces. However, it is not clear that an optimal solution algorithm for the resulting discrete equations can be developed at this stage of research,

* Department of Mathematics, University of Southern California, 1042 W. 36th Place, DRB 155, Los Angeles, CA 90089-1113. *email*: zcai@math.usc.edu

† Program in Applied Mathematics, Campus Box 526, University of Colorado at Boulder, Boulder, CO 80309-0526. This work was sponsored by the Air Force Office of Scientific Research under grant number AFOSR F49620-92-J-0439, the National Science Foundation under grant number DMS-8704169, and the Department of Energy under grant number DE-FG03-93ER25217.

‡ Program in Applied Mathematics, Campus Box 526, University of Colorado at Boulder, Boulder, CO 80309-0526. This work was sponsored by the Air Force Office of Scientific Research under grant number AFOSR-86-0126, the National Science Foundation under grant number DMS-8704169, and the Department of Energy under grant number DE-FG03-93ER25165.

albeit the matrix is symmetric and positive definite.

In this paper, we consider a least-squares functional similar to that in [5] with $q = -1$, but weighted appropriately by the Reynolds number. This is designed for the vorticity formulation of the pressure-perturbed variant of the generalized Stokes equation (which includes linear elasticity) with Dirichlet boundary conditions in two and three dimensions. Instead of applying ADN theory, we directly establish ellipticity and continuity of the functional in a product norm involving Re and the L^2- and H^1-norms. The H^{-1}-norm in the functional is further replaced by the discrete H^{-1}-norm to make the computation feasible, following the discrete H^{-1} least-squares approach proposed by Bramble, Lazarov, and Pasciak [3] for scalar second-order elliptic equations. Such discrete H^{-1} functionals are shown to be uniformly equivalent to the Sobolev norms weighted by the Reynolds number. This property enables us to show that standard finite element discretization error estimates are optimal with respect to the order of approximation as well as the required regularity of the solution, and that they are uniform in the Reynolds number. Moreover, the resulting discrete equations can be preconditioned by multigrid associated with velocity and by diagonal matrices associated with vorticity and pressure uniformly well with respect to the Reynolds number, the mesh size, and the number of levels.

The paper is organized as follows. Section 2.1 introduces the (generalized) Stokes equations, the vorticity formulation, and some preliminary results. We introduce the least-squares functional weighted appropriately by ν for the vorticity system, then establish its ellipticity and continuity in Section 2.2. Section 3 discusses the finite element approximation and Section 4 considers the discrete H^{-1}-norm least-squares functional and solution method for the resulting system of linear equations.

2. Formulations of Least-Squares Functionals. In this section, we describe the weighted least-squares functional for the vorticity formulation and show its ellipticity and continuity in the appropriate Hilbert spaces. In Subsection 2.1, we start by defining the (generalized) Stokes equation and its vorticity formulation; we next give some notation for Sobolev spaces, the divergence and curl related Hilbert spaces, and their norms; we then include some preliminary results of functional analysis. In Subsection 2.3, we introduce a least-squares functional weighted appropriately by the Reynolds number, then directly show its ellipticity and continuity.

2.1. The Stokes Equation and Its Vorticity Formulation. Let Ω be a bounded open domain in $\Re^d$ ($d = 2$ or 3) with Lipschitz boundary $\partial\Omega$. The pressure-perturbed form of the generalized stationary Stokes equation in dimensionless variables may be written as

$$\left\{ \begin{array}{rcll} -\nu\Delta\,\mathbf{u} + \nabla p &=& \mathbf{f}, & \text{in } \Omega, \\ \nabla\cdot\mathbf{u} + \delta p &=& 0, & \text{in } \Omega, \end{array} \right. \tag{2.1}$$

where the symbols Δ, ∇, and $\nabla\cdot$ stand for the Laplacian, gradient, and divergence operators, respectively; $\mathbf{f}$ is a given vector function; ν is reciprocal of the Reynolds number Re; $\mathbf{f}$ is a given vector function; and δ is some nonnegative constant ($\delta = 0$ for Stokes and $\delta = 1$ for linear elasticity with $\nu = \frac{\mu}{\lambda+\mu}$, where μ and λ are the (positive) Lamé constants). For more details on linear elasticity, see [6]. We consider the (generalized) Stokes equations (2.1) together with the Dirichlet velocity boundary condition

$$\mathbf{u} = \mathbf{0} \quad \text{on } \partial\Omega \tag{2.2}$$

and the mean pressure condition

$$\int_\Omega p\,dx = 0. \tag{2.3}$$

Let $\mathbf{curl} \equiv \nabla\times$ denote the curl operator. (Here and henceforth, we use notation for the case $d = 3$ and consider the special case $d = 2$ in the natural way by identifying $\Re^2$ with the (x_1, x_2)-plane in $\Re^3$. Thus, if $\mathbf{u}$ is two dimensional, then the curl of $\mathbf{u}$ means the scalar function

$$\nabla\times \mathbf{u} = \partial_1 u_2 - \partial_2 u_1$$

where u_1 and u_2 are the components of $\mathbf{u}$.) It can be easily checked that

$$\nabla\times(\nabla\times \mathbf{u}) = -\Delta\,\mathbf{u} + \nabla(\nabla\cdot\mathbf{u}). \tag{2.4}$$

(For $d = 2$, relation (2.4) is interpreted as

$$\nabla^\perp(\nabla\times \mathbf{u}) = -\Delta\,\mathbf{u} + \nabla(\nabla\cdot\mathbf{u}),$$

where $\nabla^\perp$ is the formal adjoint of $\nabla\times$ defined by

$$\nabla^\perp q = \begin{pmatrix} \partial_2 q \\ -\partial_1 q \end{pmatrix}.)$$

Introducing the vorticity variable

$$\omega = \nabla\times \mathbf{u},$$

using the identity (2.4), and remembering the "continuity" condition $\nabla\cdot\mathbf{u} + \delta p = 0$, then the generalized Stokes equation (2.1) may be rewritten in vorticity form as follows:

$$\left\{\begin{array}{rcll} \nu\nabla\times\omega + (1+\nu\delta)\nabla p &=& \mathbf{f}, & \text{in } \Omega, \\ \nabla\times\mathbf{u} - \omega &=& \mathbf{0}, & \text{in } \Omega, \\ \nabla\cdot\mathbf{u} + \delta p &=& 0, & \text{in } \Omega. \end{array}\right. \tag{2.5}$$

Next, we establish notation. We use the standard notation and definition for the Sobolev space $H^s(\Omega)^d$ for $s \geq 0$; the standard associated inner product and norm are denoted by $(\cdot,\cdot)_{s,\Omega}$ and $\|\cdot\|_{s,\Omega}$, respectively. (We supress the subscript d because dependence of the vector norms on dimension will be clear by context. We will omit the measure Ω from the inner product and norm designation when there is no risk of confusion.) For $s = 0$, $H^s(\Omega)^d$ coincides with $L^2(\Omega)^d$. In this case, the norm and inner product will be denoted by $\|\cdot\|$ and $(\cdot,\cdot)$, respectively. As usual, $H_0^s(\Omega)$ will denote the closure of $\mathcal{D}(\Omega)$ with respect to the norm $\|\cdot\|_s$ and $H^{-s}(\Omega)$ will denote its dual with norm defined by

$$\|\varphi\|_{-s} = \sup_{0\neq\phi\in H_0^s(\Omega)} \frac{<\varphi,\phi>}{\|\phi\|_s}.$$

Define the product spaces $H_0^s(\Omega)^d = \prod_{i=1}^d H_0^s(\Omega)$ and $H^{-1}(\Omega)^d = \prod_{i=1}^d H^{-1}(\Omega)$ with standard product norms. Let

$$H(\mathrm{div};\,\Omega) = \{\mathbf{v}\in L^2(\Omega)^d \,:\, \nabla\cdot\mathbf{v}\in L^2(\Omega)\}$$

and

$$H(\mathbf{curl};\,\Omega) = \{\mathbf{v} \in L^2(\Omega)^d \,:\, \nabla\times\mathbf{v} \in L^2(\Omega)^{2d-3}\},$$

which are Hilbert spaces under the respective norms

$$\|\mathbf{v}\|_{H(\mathrm{div};\,\Omega)} \equiv \left(\|\mathbf{v}\|^2 + \|\nabla\cdot\mathbf{v}\|^2\right)^{\frac{1}{2}}$$

and

$$\|\mathbf{v}\|_{H(\mathbf{curl};\,\Omega)} \equiv \left(\|\mathbf{v}\|^2 + \|\nabla\times\mathbf{v}\|^2\right)^{\frac{1}{2}}.$$

Define their subspaces

$$H_0(\mathrm{div};\,\Omega) = \{\mathbf{v} \in H(\mathrm{div};\,\Omega) \,:\, \mathbf{v}\cdot\mathbf{n} = 0 \text{ on } \partial\Omega\}$$

and

$$H_0(\mathbf{curl};\,\Omega) = \{\mathbf{v} \in H(\mathbf{curl};\,\Omega) \,:\, \gamma_\tau\mathbf{v} = 0 \text{ on } \partial\Omega\},$$

where $\gamma_\tau\mathbf{v} = \mathbf{v}\cdot\boldsymbol{\tau}$ for $d = 2$ and $\gamma_\tau\mathbf{v} = \mathbf{v}\times\mathbf{n}$ for $d = 3$, and $\mathbf{n}$ and $\boldsymbol{\tau}$ denote the respective unit vectors normal and tangent to the boundary. Finally, define the subspace $L_0^2(\Omega)^d$ of $L^2(\Omega)^d$ by

$$L_0^2(\Omega)^d = \{\mathbf{v} \in L^2(\Omega)^d \,:\, \int_\Omega v_i\,dx = 0 \text{ for } i = 1,\ldots,d\}.$$

Here and henceforth, we will use C with or without subscripts to denote a generic positive constant, possibly different at different occurrences; this positive constant is independent of the Reynolds parameter ν and other parameters introduced in this paper, but may depend on the domain Ω. The next lemma is an immediate consequence of a general result of functional analysis due to Nečas [12] (see also [8]).

LEMMA 2.1. *For any $p \in L_0^2(\Omega)$, there exists a positive constant C such that*

$$\|p\| \le C\,\|\nabla p\|_{-1}. \tag{2.6}$$

A result analogous to Green's formula also follows:

$$(\nabla\times\mathbf{z},\ \boldsymbol{\phi}) = (\mathbf{z},\ \nabla\times\boldsymbol{\phi}) - \int_{\partial\Omega}\boldsymbol{\phi}\cdot(\mathbf{z}\times\mathbf{n})\,ds \tag{2.7}$$

for $\mathbf{z} \in H(\mathbf{curl};\,\Omega)$ and $\boldsymbol{\phi} \in H^1(\Omega)^d$.

Finally, we will summarize results of Lemma 2.5 and Remark 2.7 in Chapter I of [8] that we will need in subsequent sections.

LEMMA 2.2. *For any $\mathbf{v} \in H_0(\mathrm{div};\,\Omega) \cap H_0(\mathbf{curl};\,\Omega)$, there exists a positive constant C such that*

$$\|\mathbf{v}\|_1 \le C\,(\|\nabla\cdot\mathbf{v}\| + \|\nabla\times\mathbf{v}\|). \tag{2.8}$$

2.2. Least-Squares Functional. Our least-squares functional is defined by the weighted sum of the L^2- and H^{-1}-norms of the residual equations of system (2.5):

$$(2.9)\quad G(\mathbf{u}, \omega, p; \mathbf{f}) = \|\mathbf{f} - (\nu\nabla\times\omega + (1+\nu\delta)\nabla p)\|_{-1}^2 + \nu^2\|\nabla\times\mathbf{u} - \omega\|^2 + \nu^2\|\nabla\cdot\mathbf{u} + \delta p\|^2.$$

(A similar functional without the weights of the Reynolds parameter ν for the Stokes equations was considered by Bochev and Gunzburger in [5].) The least-squares problem we consider is to minimize the above quadratic functional over $\mathbf{V} \equiv H_0^1(\Omega)^d \times L^2(\Omega)^{2d-3} \times L_0^2(\Omega)$: find $(\mathbf{u}, \omega, p) \in \mathbf{V}$ such that

$$(2.10)\qquad G(\mathbf{u}, \omega, p; \mathbf{f}) = \inf_{(\mathbf{v}, \sigma, q)\in\mathbf{V}} G(\mathbf{v}, \sigma, q; \mathbf{f}).$$

Next, we use an approach that departs from the established ADN theory (cf. [5]) to show ellipticity of the functional.

THEOREM 2.1. *For any $(\mathbf{u}, \omega, p) \in \mathbf{V}$, positive constants C_1 and C_2 exist independent of ν such that*

$$(2.11)\qquad C_1\left(\nu^2\|\mathbf{u}\|_1^2 + \nu^2\|\omega\|^2 + (1+\nu\delta)^2\|p\|^2\right) \le G(\mathbf{u}, \omega, p; \mathbf{0})$$

and

$$(2.12)\qquad G(\mathbf{u}, \omega, p; \mathbf{0}) \le C_2\left(\nu^2\|\mathbf{u}\|_1^2 + \nu^2\|\omega\|^2 + (1+\nu\delta)^2\|p\|^2\right).$$

Proof. Upper bound (2.12) is straightforward from the triangle and Cauchy-Schwarz inequalities. We proceed to show the validity of (2.11) for $(\mathbf{u}, \omega, p) \in H_0^1(\Omega)^d \times H(\mathbf{curl};\, \Omega) \times (L_0^2(\Omega) \cap H^1(\Omega))$. It will then follow for $(\mathbf{u}, \omega, p) \in \mathbf{V}$ by continuity. Now from (2.7) and the Cauchy-Schwarz inequality, for any $\phi \in H_0^1(\Omega)^d$ we have

$$\begin{aligned}\frac{1+\nu\delta}{\nu}(\nabla p, \phi) &= \frac{1}{\nu}(\nu\nabla\times\omega + (1+\nu\delta)\nabla p, \phi) + (\nabla\times\mathbf{u} - \omega, \nabla\times\phi) - (\nabla\times\mathbf{u}, \nabla\times\phi)\\ &\le C\left(\frac{1}{\nu}\|\nu\nabla\times\omega + (1+\nu\delta)\nabla p\|_{-1} + \|\nabla\times\mathbf{u} - \omega\| + \|\nabla\times\mathbf{u}\|\right)\|\phi\|_1,\end{aligned}$$

which, together with Lemma 2.1, implies that

$$\begin{aligned}\frac{1+\nu\delta}{\nu}\|p\| &\le C\frac{1+\nu\delta}{\nu}\|\nabla p\|_{-1}\\ &\le C\left(\frac{1}{\nu}\|\nu\nabla\times\omega + (1+\nu\delta)\nabla p\|_{-1} + \|\nabla\times\mathbf{u} - \omega\| + \|\nabla\times\mathbf{u}\|\right). \qquad (2.13)\end{aligned}$$

By (2.7), the Cauchy-Schwarz and triangle inequalities, Lemma 2.2, and (2.13), we have that

$$\begin{aligned}\|\nabla\times\mathbf{u}\|^2 &= (\nabla\times\mathbf{u} - \omega, \nabla\times\mathbf{u}) + \frac{1}{\nu}(\nu\nabla\times\omega + (1+\nu\delta)\nabla p, \mathbf{u})\\ &\quad + \frac{1+\nu\delta}{\nu}(p, \nabla\cdot\mathbf{u} + \delta p) - \frac{\delta(1+\nu\delta)}{\nu}(p, p)\\ &\le \|\nabla\times\mathbf{u} - \omega\|\,\|\nabla\times\mathbf{u}\| + \frac{1}{\nu}\|\nu\nabla\times\omega + (1+\nu\delta)\nabla p\|_{-1}\,\|\mathbf{u}\|_1\\ &\quad + \frac{1+\nu\delta}{\nu}\|p\|\,\|\nabla\cdot\mathbf{u} + \delta p\|\end{aligned}$$

$$
\begin{aligned}
&\leq \frac{C}{\nu}\|\nu\nabla\times\omega + (1+\nu\delta)\nabla p\|_{-1}\,(\|\nabla\times\mathbf{u}\| + \|\nabla\cdot\mathbf{u}+\delta p\| + \delta\|p\|)\\
&\qquad + \|\nabla\times\mathbf{u}-\omega\|\,\|\nabla\times\mathbf{u}\| + \frac{1+\nu\delta}{\nu}\|p\|\,\|\nabla\cdot\mathbf{u}+\delta p\|\\
&\leq \|\nabla\times\mathbf{u}\|\left(\|\nabla\times\mathbf{u}-\omega\| + \frac{C}{\nu}\|\nu\nabla\times\omega + (1+\nu\delta)\nabla p\|_{-1} + \|\nabla\cdot\mathbf{u}+\delta p\|\right)\\
&\qquad + \frac{C}{\nu^2}G(\mathbf{u},\omega,p;\mathbf{0})\\
&\leq \frac{1}{2}\|\nabla\times\mathbf{u}\|^2 + \frac{C}{\nu^2}G(\mathbf{u},\omega,p;\mathbf{0}).
\end{aligned}
$$

Hence,

$$
\|\nabla\times\mathbf{u}\|^2 \leq \frac{C}{\nu^2}G(\mathbf{u},\omega,p;\mathbf{0}). \tag{2.14}
$$

But (2.14), (2.13), the bounds

$$
\|\omega\| \leq \|\nabla\times\mathbf{u}-\omega\| + \|\nabla\times\mathbf{u}\| \quad\text{and}\quad \|\nabla\cdot\mathbf{u}\| \leq \|\nabla\cdot\mathbf{u}+\delta p\| + \delta\,\|p\|,
$$

and Lemma 2.2 imply (2.11). This completes the proof of the theorem. □

3. Finite Element Approximations. We approximate the minimum of $G(\mathbf{u},\omega,p;\mathbf{f})$ in (2.10) using a Rayleigh-Ritz type finite element method. Assuming that the domain Ω is a polyhedron, let $\mathcal{T}_h$ be a partition of the Ω into finite elements, i.e., $\Omega = \cup_{K\in\mathcal{T}_h}K$ with $h = \max\{\mathrm{diam}(K) : K\in\mathcal{T}_h\}$. Assume that the triangulation $\mathcal{T}_h$ is quasi-uniform, i.e., it is regular and satisfies the inverse assumption (see [7]). Let $\mathbf{V}_h = \mathbf{U}_h\times\mathbf{W}_h\times P_h$ be a finite-dimensional subspace of $\mathbf{V}$ with the following properties: for any $(\mathbf{u},\omega,p)\in\left(H^{r+1}(\Omega)^d\times H^r(\Omega)^{2d-2}\right)\cap\mathbf{V}$,

$$
\inf_{\mathbf{v}\in\mathbf{U}_h}(\|\mathbf{u}-\mathbf{v}\| + h\,\|\mathbf{u}-\mathbf{v}\|_1) \leq Ch^{r+1}\|\mathbf{u}\|_{r+1}, \tag{3.1}
$$

$$
\inf_{\sigma\in\mathbf{W}_h}(\|\omega-\sigma\| + h\,\|\omega-\sigma\|_1) \leq Ch^{r}\|\omega\|_{r}, \tag{3.2}
$$

$$
\inf_{q\in P_h}(\|p-q\| + h\,\|p-q\|_1) \leq Ch^{r}\|p\|_{r}. \tag{3.3}
$$

where $r\geq 1$ is integer. It is well-known that (3.1)–(3.3) holds for typical finite element spaces consisting of piecewise polynomials with respect to quasi-uniform triangulations (cf. [7]).

The finite element approximation to minimizing $G(\mathbf{u},\omega,p;\mathbf{f})$ in (2.10) on $\mathbf{V}$ becomes: find $(\mathbf{u}_h,\omega_h,p_h)\in\mathbf{V}_h$ that satisfies

$$
G(\mathbf{u}_h,\omega_h,p_h;\mathbf{f}) = \inf_{(\mathbf{v},\sigma,q)\in\mathbf{V}_h} G(\mathbf{v},\sigma,q;\mathbf{f}). \tag{3.4}
$$

Denote the norm induced by the functional according to

$$
|||(\mathbf{u},\omega,p)||| \equiv \left(\nu^2\|\mathbf{u}\|_1^2 + \nu^2\|\omega\|^2 + (1+\nu\delta)^2\|p\|^2\right)^{\frac{1}{2}}.
$$

THEOREM 3.1. *Assume that* $(\mathbf{u},\omega,p)\in\left(H^{r+1}(\Omega)^d\times H^r(\Omega)^{2d-2}\right)\cap\mathbf{V}$ *is the solution of problem* (2.10). *Then*

$$
|||(\mathbf{u},\omega,p)-(\mathbf{u}_h,\omega_h,p_h)|||_{\mathbf{V}} \leq C\,h^r d_r(\mathbf{u},\omega,p), \tag{3.5}
$$

where C depends only on the domain Ω and the ratio of the constants C_2 and C_1 in Theorem 2.1 *and where*

$$d_r(\mathbf{u}, \omega, p) = \left(\nu^2 \|\mathbf{u}\|_{r+1}^2 + \nu^2 \|\omega\|_r^2 + (1+\nu\delta)^2 \|p\|_r^2\right)^{\frac{1}{2}}. \tag{3.6}$$

Proof. It is easy to see that the error $(\mathbf{u}-\mathbf{u}_h, \omega-\omega_h, p-p_h)$ is orthogonal to $\mathbf{V}_h$ with respect to the inner product corresponding to the norm $|||\cdot|||_{\mathbf{V}}$. Bound (3.5) now follows from Theorem 2.1 and approximation properties (3.1)–(3.3). □

REMARK 3.1. *The above result indicates that the finite element approximation is optimal, both with respect to the order of approximation and the required regularity of the solution (see* [3]*). More specifically, bound* (3.5) *holds with*

$$d_r(\mathbf{u}, \omega, p) = \left(\nu^2 \|\mathbf{u}\|_{r+1}^2 + \|p\|_r^2\right)^{\frac{1}{2}}$$

since $\omega = \nabla \times \mathbf{u}$ and $\|\nabla \times \mathbf{u}\|_r \leq C \|\mathbf{u}\|_{r+1}$.

4. Solution Method and Discrete H^{-1} Functional. Theorem 3.1 indicates that the finite element approximation based on the functional G is also optimal with respect to the required regularity of the solution. Notice that the functional involves the H^{-1} norm, which in turn requires solution of a boundary value problem for its evaluation. There are two existing approaches to make the method computationally feasible: the mesh-dependent least-squares scheme proposed by Aziz, Kellogg, and Stephens [2] (see also [5]) and the discrete H^{-1}-norm scheme proposed by Bramble, Lazarov, and Pasciak [3]. As mentioned in the introduction, it is not clear that a fast solution algorithm for the resulting discrete equations from the mesh-dependent least-squares method can be developed at this stage of research. In this paper, we will therefore adopt the discrete H^{-1}-norm approach. Following [3], the H^{-1}-norm in the functional is replaced by a discrete norm. This discrete H^{-1} functional is computable and can be uniformly preconditioned by well-known techniques.

To this end, let $A : H^{-1}(\Omega)^d \longrightarrow H_0^1(\Omega)^d$ denote the solution operator for the Poisson problem

$$\begin{cases} -\Delta \phi &= \mathbf{v}, \quad \text{in} \quad \Omega, \\ \phi &= 0, \quad \text{on} \quad \partial\Omega, \end{cases} \tag{4.1}$$

i.e., $A\mathbf{v} = \phi$ for a given $\mathbf{v} \in H^{-1}(\Omega)^d$ is the solution to (4.1). It is well-known that $\sqrt{(A\cdot, \cdot)}$ defines a norm that is equivalent to the $H^{-1}(\Omega)^d$ norm. Let $A_h : L^2(\Omega)^d \longrightarrow \mathbf{U}_h$ be defined by $A_h\varphi = \phi$, where ϕ is the unique solution in $\mathbf{U}_h$ satisfying

$$\int_\Omega \nabla \phi \cdot \nabla \psi \, dx = (\varphi, \psi), \quad \forall\, \psi \in \mathbf{U}_h.$$

Assume that there is a preconditioner $B_h : L^2(\Omega)^d \longrightarrow \mathbf{U}_h$ that is symmetric with respect to the $L^2(\Omega)$ inner product and spectrally equivalent to A_h, i.e., there are positive constants C_1 and C_2, not depending on h, that satisfy

$$C_1 (A_h\phi, \phi) \leq (B_h\phi, \phi) \leq C_2 (A_h\phi, \phi), \quad \forall\, \phi \in \mathbf{U}_h. \tag{4.2}$$

Following [3], define $\tilde{A}_h = h^2 I + B_h$ where I denotes the identity operator on $\mathbf{U}_h$. In the remainder of this section, we analyze the least-squares approximation based on the

functional

$$G^h(\mathbf{u}, \boldsymbol{\omega}, p; \mathbf{f}) = \left(\tilde{A}_h(\mathbf{f} - (\nu\nabla\times\boldsymbol{\omega} + (1+\nu\delta)\nabla p)), \mathbf{f} - (\nu\nabla\times\boldsymbol{\omega} + (1+\nu\delta)\nabla p)\right)$$

$$+ \nu^2 \|\nabla\times\mathbf{u} - \boldsymbol{\omega}\|^2 + \nu^2 \|\nabla\cdot\mathbf{u} + \delta p\|^2. \tag{4.3}$$

Define the norm corresponding to the functional G^h by

$$|||(\mathbf{u}, \boldsymbol{\omega}, p)|||_{\mathbf{V}_h} \equiv \sqrt{G^h(\mathbf{u}, \boldsymbol{\omega}, p; \mathbf{0})}.$$

Let $Q_h : L^2(\Omega)^d \longrightarrow \mathbf{U}_h$ denote the $L^2(\Omega)^d$ orthogonal projection operator onto $\mathbf{U}_h$. We assume that Q_h is bounded on $H^1(\Omega)^d$, i.e.,

$$\|Q_h\mathbf{v}\|_1 \le C\,\|\mathbf{v}\|_1, \quad \forall\, \mathbf{v} \in H^1(\Omega)^d. \tag{4.4}$$

REMARK 4.1. *The symmetry of B_h with respect to the inner product on $L^2(\Omega)^d$ implies that $B_h = B_hQ_h$. Similarly, $A_h = A_hQ_h$. Thus, (4.2) holds for any $\mathbf{v} \in L^2(\Omega)^d$.*

It is easy to check that assumptions (3.1) and (4.4) imply that

$$\|(I - Q_h)\mathbf{v}\|_{-1} \le Ch\,\|\mathbf{v}\|, \quad \forall\, \mathbf{v} \in L^2(\Omega)^d, \tag{4.5}$$

and that (see [3])

$$\|Q_h\mathbf{v}\|_{-1}^2 \le C\,(A_h\mathbf{v}, \mathbf{v}) \le C\,\|\mathbf{v}\|_{-1}^2, \quad \forall\, \mathbf{v} \in L^2(\Omega)^d. \tag{4.6}$$

LEMMA 4.1. *For any $(\mathbf{u}, \boldsymbol{\omega}, p) \in H_0^1(\Omega)^d \times H(\mathbf{curl}; \Omega) \times (L_0^2(\Omega) \cap H^1(\Omega))$, positive constants C_1 and C_2 exist, independent of h and ν, such that*

$$C_1 \left(\nu^2\|\mathbf{u}\|_1^2 + \nu^2\|\boldsymbol{\omega}\|^2 + (1+\nu\delta)^2\|p\|^2\right) \le |||(\mathbf{u}, \boldsymbol{\omega}, p)|||_{\mathbf{V}_h}^2$$

$$\le C_2 \left(\nu^2\|\mathbf{u}\|_1^2 + \nu^2h^2\|\nabla\times\boldsymbol{\omega}\|^2 + \nu^2\,\|\boldsymbol{\omega}\|^2 + h^2(1+\nu\delta)^2\|\nabla p\|^2 + (1+\nu\delta)^2\|p\|^2\right). \tag{4.7}$$

Proof. By Remark 4.1 and (4.6), we have that

$$(\tilde{A}_h\boldsymbol{\phi}, \boldsymbol{\phi}) \le C\left(h^2\|\boldsymbol{\phi}\|^2 + (A_h\boldsymbol{\phi}, \boldsymbol{\phi})\right) \le C\left(h^2\|\boldsymbol{\phi}\|^2 + \|\boldsymbol{\phi}\|_{-1}^2\right), \quad \forall\, \boldsymbol{\phi} \in L^2(\Omega)^d,$$

which, together with the triangle inequality and Theorem 2.1, imply the upper bound in (4.7). To prove the first inequality in (4.7), by Theorem 2.1 it suffices to show that

$$\|\nu\nabla\times\boldsymbol{\omega} + (1+\nu\delta)\nabla p\|_{-1}^2 \le C\left(\tilde{A}_h(\nu\nabla\times\boldsymbol{\omega} + (1+\nu\delta)\nabla p), \nu\nabla\times\boldsymbol{\omega} + (1+\nu\delta)\nabla p\right)$$

for any $\boldsymbol{\omega} \in H(\mathbf{curl}; \Omega)$ and any $p \in H^1(\Omega)$. From (4.5), (4.6), and Remark 4.1, for any $\boldsymbol{\phi} \in L^2(\Omega)^d$ we have

$$\begin{aligned}\|\boldsymbol{\phi}\|_{-1}^2 &\le 2\left(\|(I - Q_h)\boldsymbol{\phi}\|_{-1}^2 + \|Q_h\boldsymbol{\phi}\|_{-1}^2\right)\\ &\le C\left(h^2\|\boldsymbol{\phi}\|^2 + (A_h\boldsymbol{\phi}, \boldsymbol{\phi})\right)\\ &\le C\,(\tilde{A}_h\boldsymbol{\phi}, \boldsymbol{\phi}).\end{aligned}$$

This completes the proof of the lemma. □

REMARK 4.2. *If* $\mathbf{W}_h \subset H(\mathbf{curl};\Omega)$ *and* $P_h \subset L_0^2(\Omega)\cap H^1(\Omega)$ *satisfy an inverse inequality of the form*

$$\|\nabla\times\boldsymbol{\omega}\| \le C\,h^{-1}\|\boldsymbol{\omega}\| \quad \textit{and} \quad \|\nabla p\| \le C\,h^{-1}\|p\|,$$

respectively, then the second inequality of (4.7) *can be replaced by* $\nu^2\|\mathbf{u}\|_1^2+\nu^2\,\|\boldsymbol{\omega}\|^2+(1+\nu\delta)^2\|p\|^2$ *for any* $\mathbf{u}\in H_0^1(\Omega)^d$, *any* $\boldsymbol{\omega}\in\mathbf{W}_h$, *and any* $p\in P_h$. *It is well-known (cf. [7]) that the above inverse inequalities hold for typical finite element spaces consisting of piecewise polynomials on quasi-uniform triangulations.*

THEOREM 4.1. *Let* $(\mathbf{u}_h,\boldsymbol{\omega}_h,p_h)\in\mathbf{V}_h$ *be the unique minimizer of* $G^h(\mathbf{u},\boldsymbol{\omega},p;\mathbf{f})$ *over* $\mathbf{V}_h$ *and let* $(\mathbf{u},\boldsymbol{\omega},p)\in\left(H^{r+1}(\Omega)^d\times H^r(\Omega)^d\times H^r(\Omega)\right)\cap\mathbf{V}$ *be the solution of problem* (2.10). *Then*

$$(4.8)\quad \nu\,\|\mathbf{u}-\mathbf{u}_h\|_1+\nu\,\|\boldsymbol{\omega}-\boldsymbol{\omega}_h\|+(1+\nu\delta)\|p-p_h\| \le C\,h^r\left(\nu^2\|\mathbf{u}\|_{r+1}^2+(1+\nu\delta)^2\|p\|_r^2\right)^{\frac{1}{2}},$$

where C *is independent of the mesh size* h *and the Reynolds parameter* ν.

Proof. It is easy to see that the error $(\mathbf{u}-\mathbf{u}_h,\boldsymbol{\omega}-\boldsymbol{\omega}_h,p-p_h)$ is orthogonal to $\mathbf{V}_h$ with respect to the inner product corresponding to the norm $|||\cdot|||_{\mathbf{V}_h}$. Bound (4.8) now follows from Lemma 4.1 and approximation properties (3.1)–(3.2). □

For the finite element spaces $\mathbf{W}_h$ and P_h satisfying the inverse inequalities in Remark 4.2, the discrete H^{-1} functional $G^h(\mathbf{u},\boldsymbol{\omega},p;\mathbf{0})$ can be preconditioned by the functional $\nu^2\|\mathbf{u}\|_1^2+\nu^2\,\|\boldsymbol{\omega}\|^2+(1+\nu\delta)^2\|p\|^2$ that decouples velocity, vorticity, and pressure unknowns, because they are uniformly spectral equivalent in the mesh size h and the Reynolds parameter ν (see Lemma 4.1 and Remark 4.2). We can use any effective elliptic preconditioners associated with velocity $\mathbf{u}$, including those of multigrid type, and simple preconditioners associated with vorticity $\boldsymbol{\omega}$ and pressure p, including those of diagonal matrix type.

Acknowledgments. We thank Professors Pavel Bochev and Seymour Parter for helpful discussions.

REFERENCES

[1] S. AGMON, A. DOUGLIS, AND L. NIRENBERG, *Estimates near the boundary for solutions of elliptic partial differential equations satisfying general boundary conditions II*, Comm. Pure Appl. Math., 17 (1964), pp. 35–92.

[2] A. K. AZIZ, R. B. KELLOGG, AND A.B. STEPHENS, *Least-squares methods for elliptic systems*, Math. Comp., 44:169 (1985), pp. 53–70.

[3] J. H. BRAMBLE, R. D. LAZAROV, AND J. E. PASCIAK, *A least-squares approach based on a discrete minus one inner product for first order system*, Manuscript.

[4] P. B. BOCHEV AND M. D. GUNZBURGER, *Accuracy of least-squares methods for the Navier–Stokes equations*, Comput. Fluids, 22 (1993), pp. 549–563.

[5] P. B. BOCHEV AND M. D. GUNZBURGER, *Analysis of least-squares finite element methods for the Stokes equations*, Math. Comp., 63:208 (1994), pp. 479–506.

[6] Z. CAI, T. MANTEUFFEL, AND S. MCCORMICK, *First-order system least squares for the Stokes equations*, SIAM J. Numer. Anal., submitted.

[7] P. G. CIARLET, *The Finite Element Method for Elliptic Problems*, North-Holland, New York, 1978.

[8] V. GIRAULT AND P. A. RAVIART, *Finite Element Methods for Navier-Stokes Equations: Theory and Algorithms*, Springer-Verlag, New York, 1986.

[9] B. N. JIANG AND C. CHANG, *Least-squares finite elements for the Stokes problem*, Comput. Meth. Appl. Mech. Engrg., 78 (1990), pp. 297–311.

[10] B. N. JIANG AND L. POVINELLI, *Least-squares finite element method for fluid dynamics*, Comput. Meth. Appl. Mech. Engrg., 81 (1990), pp. 13–37.

[11] B. N. JIANG AND V. SONNAD, *Least-squares solution of incompressible Navier-Stokes equations with the p-version of finite elements*, NASA TM 105203 (ICOMP Rep. 91-14), 1991.

[12] J. NEČAS, *Equations aux Dérivées Partielles*, Presses de l'Université de Montréal, 1965.

[13] L. TANG AND T. TSANG, *A least-squares finite element method for time-dependent incompressible flows with thermal convection*, Inter. J. Numer. Meth. Fluids, to appear.

FIRST-ORDER SYSTEM LEAST SQUARES FOR THE STOKES EQUATIONS, WITH APPLICATION TO LINEAR ELASTICITY

Z. CAI*, T. A. MANTEUFFEL†, AND S. F. MCCORMICK†

Abstract. Following our earlier work on general second-order scalar equations, here we develop a least-squares functional for the two- and three-dimensional Stokes equations, generalized slightly by allowing a pressure term in the continuity equation. By introducing a *velocity flux* variable and associated curl and *trace* equations, we are able to establish ellipticity in an H^1 product norm appropriately weighted by the Reynolds number. This immediately yields optimal discretization error estimates for finite element spaces in this norm and optimal algebraic convergence estimates for multiplicative and additive multigrid methods applied to the resulting discrete systems. Both estimates are uniform in the Reynolds number. Moreover, our pressure-perturbed form of the generalized Stokes equations allows us to develop an analogous result for the Dirichlet problem for linear elasticity with estimates that are uniform in the Lamé constants.

Key words. least squares, multigrid, Stokes equations

AMS(MOS) subject classifications. 65F10, 65F30

1. Introduction.

In earlier work [9, 10], we developed least-squares functionals for a first-order system formulation of general second-order elliptic scalar partial differential equations. The functional developed in [10] was shown to be elliptic in the sense that its homogeneous form applied to the $n+1$ variables (*pressure* and *velocities*) is equivalent to the $(H^1)^{n+1}$ norm. This means that the individual variables in the functional are essentially decoupled (more precisely, their interactions are essentially subdominant). This important property ensures that standard finite element methods are of H^1-optimal accuracy in each variable and that multiplicative and additive multigrid methods applied to the resulting discrete equations are optimally convergent.

The purpose of this paper is to extend this methodology to the Stokes equations in two and three dimensions. To this end, we begin by reformulating the Stokes equations as a first-order system derived in terms of an additional vector variable, the *velocity flux*, defined as the vector of gradients of the Stokes velocities. We first apply a least-squares principle to this system using L^2 and H^{-1} norms weighted appropriately by the Reynolds number, Re. We then show that the resulting functional is elliptic in a product norm involving Re and the L^2 and H^1 norms. While of theoretical interest in its own right, we use this result here primarily as a vehicle for establishing that a modified form of this functional is fully elliptic in an H^1 product norm scaled by Re.

This appears to be the first general theory of this kind for the Stokes equations in general dimensions with velocity boundary conditions. Bochev and Gunzburger [6] developed least-squares functionals for Stokes equations in norms that include stronger Sobolev terms and mesh weighting, but none are product H^1 elliptic. Chang [11] also used velocity derivative variables to derive a product H^1 elliptic functional for Stokes equations, but it is inherently limited to two dimensions. For general dimensions, a vorticity-velocity-pressure form (cf.

* Department of Mathematics, University of Southern California, 1042 W. 36th Place, DRB 155, Los Angeles, CA 90089-1113. *email*: zcai@math.usc.edu.

† Program in Applied Mathematics, Campus Box 526, University of Colorado at Boulder, Boulder, CO 80309-0526. *email*: tmanteuf@boulder.colorado.edu and stevem@boulder.colorado.edu. This work was sponsored by the Air Force Office of Scientific Research under grant number AFOSR-91-0156, the National Science Foundation under grant number DMS-8704169, and the Department of Energy under grant number DE-FG03-93ER25165.

[4, 15]) proved to be product H^1 elliptic, but only for certain nonstandard boundary conditions. For the more practical (cf. [14, 17, 19]) velocity boundary conditions treated here, the velocity-vorticity-pressure formulation examined by Chang [12] can be shown by counterexample [3] not to be equivalent to any H^1 product norm, even with the added boundary condition on the normal component of vorticity. Moreover, this formulation admits no apparent additional equation, such as the curl and trace constraints introduced below for our formulation, which would enable such an equivalence. The velocity-pressure-stress formulation described in [7] has the same shortcomings. (If the vorticity and deformation stress variables are important, then they can be easily and accurately reconstructed from the velocity-flux variables introduced in our formulation.)

While our least-squares form requires several new dependent variables, we believe that the added cost is more than offset by the strengthened accuracy of the discretization and the speed that the attendant multigrid solution process attains. Moreover, our modified functional requires strong regularity conditions; this requirement is to be expected for obtaining full product H^1 ellipticity in all variables, including velocity fluxes. (We thus obtain optimal H^1 estimates for the derivatives of velocity.) In any case, strengthened regularity is not necessary for the first functional we introduce.

Our modified Stokes functional is obtained essentially by augmenting the first-order system with a curl constraint and a scalar (*trace*) equation involving certain derivatives of the velocity flux variable and then appealing to a simple L^2 least-squares principle. As in [10] for the scalar case, the important H^1 ellipticity property that we establish guarantees optimal finite element accuracy and multigrid convergence rates applied to this Stokes least-squares functional that are uniform in Re.

One of the more compelling benefits of least squares is the freedom to incorporate additional equations and impose additional boundary conditions as long as the system is consistent. In fact, many problems are perhaps best treated with overdetermined (but consistent) first-order systems, as we have here for the Stokes equations. We therefore abandon the so-called ADN theory (cf. [1, 2]), which is restricted to square systems, in favor of more direct tools of analysis.

An important aspect of our general formulation is that it applies equally well to the Dirichlet problem for linear elasticity. This is done by posing the Stokes equations in a slightly generalized form that includes a pressure term in the continuity equation. Our development and results then automatically apply to linear elasticity. Most important, our optimal discretization and solver estimates are uniform in the Lamé constants.

We emphasize that the discretization and algebraic convergence properties for the generalized Stokes equations are automatic consequences of the H^1 product norm ellipticity established here and the finite element and multigrid theories established in Sections 3-5 of [10]. We are therefore content with an abbreviated paper that focuses on establishing ellipticity, which we do in Section 3. Section 2 introduces the generalized Stokes equations, the two relevant first-order systems and their functionals, and some preliminary theory. Concluding remarks are made in Section 4.

2. The Stokes Problem, Its First-Order System Formulation, and Other Preliminaries. Let Ω be a bounded, open, connected domain in $\Re^n$ ($n = 2$ or 3) with Lipschitz boundary $\partial\Omega$. The pressure-perturbed form of the generalized stationary Stokes

equations in dimensionless variables may be written as

$$\text{(2.1)} \qquad \begin{cases} -\nu\Delta\,\mathbf{u} + \nabla p &= \mathbf{f}, & \text{in} \quad \Omega, \\ \nabla\cdot\,\mathbf{u} + \delta p &= g, & \text{in} \quad \Omega, \end{cases}$$

where the symbols Δ, ∇, and $\nabla\cdot$ stand for the Laplacian, gradient, and divergence operators, respectively; ν is the reciprocal of the Reynolds number Re; $\mathbf{f}$ is a given vector function; g is a given scalar function; and δ is some nonnegative constant ($\delta = 0$ for Stokes, $\delta = 1$ for linear elasticity). Without loss of generality, we may assume that

$$\text{(2.2)} \qquad \int_\Omega g\,dz = \int_\Omega p\,dz = 0.$$

(For $\delta = 0$, equation (2.1) can have a solution only when g satisfies (2.2), and we are then free to ask that p satisfy (2.2). For $\delta > 0$, in general we have only that $\int_\Omega g\,dz = \delta \int_\Omega p\,dz$, but this can be reduced to (2.2) simply by replacing p by $p - \frac{g}{\delta}$ and g by 0 in (2.1).) We consider the (generalized) Stokes equations (2.1) together with the Dirichlet velocity boundary condition

$$\text{(2.3)} \qquad \mathbf{u} = \mathbf{0} \quad \text{on } \partial\Omega.$$

The slightly generalized Stokes equations in (2.1) allow our results to apply to linear elasticity. In particular, consider the Dirichlet problem

$$\text{(2.4)} \qquad \begin{cases} -\mu\Delta\,\mathbf{u} - (\lambda+\mu)\nabla\nabla\cdot\,\mathbf{u} &= \mathbf{f}, & \text{in} \quad \Omega, \\ \mathbf{u} &= \mathbf{0}, & \text{on} \quad \partial\Omega, \end{cases}$$

where $\mathbf{u}$ now represents displacements and μ and λ are the (positive) Lamé constants. By $\Delta\,\mathbf{u}$ here we mean the n-vector of components $\Delta\,u_i$; that is, Δ applies to $\mathbf{u}$ componentwise. This is recast in form (2.1)-(2.2) by introducing the pressure variable[1] $p = -\nabla\cdot\,\mathbf{u}$, by rescaling $\mathbf{f}$, and by letting $g = 0$, $\delta = 1$, and $\nu = \frac{\mu}{\lambda+\mu}$. (It is easy to see that this p must satisfy (2.2).) An important consequence of the results we develop below is that standard Rayleigh-Ritz discretization and multigrid solution methods can be applied with optimal estimates that are uniform in h, λ, and μ. For example, we obtain optimal uniform approximation of the gradients of displacements in the H^1 product norm. This in turn implies analogous H^1 estimates for the stresses, which are easily obtained from the "velocity fluxes". For related results with a different methodology and weaker norm estimates, see [13].

Let $\mathbf{curl} \equiv \nabla\times$ denote the curl operator. (Here and henceforth, we use notation for the case $n = 3$ and consider the special case $n = 2$ in the natural way by identifying $\Re^2$ with the (x_1, x_2) plane in $\Re^3$. Thus, if $\mathbf{u}$ is two dimensional, then the curl of $\mathbf{u}$ means the scalar function

$$\nabla\times\,\mathbf{u} = \partial_1 u_2 - \partial_2 u_1,$$

[1] Perhaps a more physical choice for this artificial pressure would have been $p = -\frac{\lambda}{2\mu}\nabla\cdot\,\mathbf{u}$, since it then becomes the hydrostatic pressure in the incompressible limit. We chose our particular scaling because it most easily conforms to (2.1). In any case, our results apply to virtually any nonnegative scaling of p, with no effect on the equivalence constants (provided the norms are correspondingly scaled); see Theorems 3.1 and 3.2.

where u_1 and u_2 are the components of $\mathbf{u}$.) The following identity is immediate:

$$\nabla\times(\nabla\times\mathbf{u}) = -\Delta\,\mathbf{u} + \nabla(\nabla\cdot\mathbf{u}). \tag{2.5}$$

(For $n = 2$, (2.5) is interpreted as

$$\nabla^{\perp}(\nabla\times\mathbf{u}) = -\Delta\,\mathbf{u} + \nabla(\nabla\cdot\mathbf{u}),$$

where $\nabla^{\perp}$ is the formal adjoint of $\nabla\times$ defined by

$$\nabla^{\perp}q = \begin{pmatrix} \partial_2 q \\ -\partial_1 q \end{pmatrix}.)$$

We will be introducing a new independent variable defined as the n^2-vector function of gradients of the u_i, $i = 1, 2, ..., n$. It will be convenient to view the original n-vector functions as column vectors and the new n^2-vector functions as either block column vectors or matrices. Thus, given

$$\mathbf{u} = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{pmatrix}$$

and denoting $\mathbf{u}^t = (u_1, u_2, ..., u_n)$, then an operator G defined on scalar functions (e.g., $G = \nabla$) is extended to n-vectors componentwise:

$$G\mathbf{u}^t = (Gu_1, Gu_2, ..., Gu_n)$$

and

$$G\mathbf{u} = \begin{pmatrix} Gu_1 \\ Gu_2 \\ \vdots \\ Gu_n \end{pmatrix}.$$

If $\mathbf{U}_i \equiv G\mathbf{u}_i$ is a n-vector function, then we write the matrix

$$\begin{aligned} \underline{\mathbf{U}} \equiv G\mathbf{u}^t &= (\mathbf{U}_1, \mathbf{U}_2, ..., \mathbf{U}_n) \\ &= \begin{pmatrix} U_{11} & U_{12} & \cdots & U_{1n} \\ U_{21} & U_{22} & \cdots & U_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ U_{n1} & U_{n2} & \cdots & U_{nn} \end{pmatrix}. \end{aligned}$$

We then define the *trace* operator tr according to

$$tr\underline{\mathbf{U}} = \sum_{i=1}^{n} U_{ii}.$$

If D is an operator on n-vector functions (e.g., $D = \nabla\times$), then its extension to matrices is defined by

$$D\underline{\mathbf{U}} = (D\mathbf{U}_1, D\mathbf{U}_2, \cdots, D\mathbf{U}_n).$$

When each $D\mathbf{U}_i$ is a scalar function (e.g., $D = \nabla\cdot$), then we will want to view the extension as a mapping to column vectors, so we will use the convention

$$(D\underline{\mathbf{U}})^t = \begin{pmatrix} D\mathbf{U}_1 \\ D\mathbf{U}_2 \\ \vdots \\ D\mathbf{U}_n \end{pmatrix}.$$

We also extend the tangential operator $\mathbf{n}\times$ componentwise:

$$\mathbf{n}\times\underline{\mathbf{U}} = (\mathbf{n}\times\mathbf{U}_1,\ \mathbf{n}\times\mathbf{U}_2,\ \cdots,\ \mathbf{n}\times\mathbf{U}_n).$$

Finally, inner products and norms on the matrix functions are defined in the natural componentwise way, e.g.,

$$\|\underline{\mathbf{U}}\|^2 = \sum_{i=1}^{n}\|\mathbf{U}_i\|^2 = \sum_{i,j=1}^{n}\|U_{ij}\|^2.$$

If we introduce the *velocity flux* variable

$$\underline{\mathbf{U}} = \nabla\mathbf{u}^t = (\nabla u_1,\ \nabla u_2,\ \cdots,\ \nabla u_n),$$

then the Stokes system (2.1) and (2.3) may be recast as the following equivalent first-order system:

$$\text{(2.6)}\qquad \left\{\begin{array}{rcll} \underline{\mathbf{U}} - \nabla\mathbf{u}^t &=& \underline{\mathbf{0}}, & \text{in } \Omega,\\ -\nu\,(\nabla\cdot\underline{\mathbf{U}})^t + \nabla p &=& \mathbf{f}, & \text{in } \Omega,\\ \nabla\cdot\mathbf{u} + \delta p &=& g, & \text{in } \Omega,\\ \mathbf{u} &=& \mathbf{0}, & \text{on } \partial\Omega. \end{array}\right.$$

Note that the definition of $\underline{\mathbf{U}}$, the "continuity" condition $\nabla\cdot\mathbf{u} + \delta p = g$ in Ω, and the Dirichlet condition $\mathbf{u} = \mathbf{0}$ on $\partial\Omega$ imply the respective properties

$$\text{(2.7)}\qquad \nabla\times\underline{\mathbf{U}} = \underline{\mathbf{0}} \ \text{ in } \Omega, \quad tr\,\underline{\mathbf{U}} + \delta p = g \ \text{ in } \Omega, \quad \text{and } \mathbf{n}\times\underline{\mathbf{U}} = \underline{\mathbf{0}} \ \text{ on } \partial\Omega.$$

Then an equivalent extended system for (2.6) is

$$\text{(2.8)}\qquad \left\{\begin{array}{rcll} \underline{\mathbf{U}} - \nabla\mathbf{u}^t &=& \mathbf{0}, & \text{in } \Omega,\\ -\nu\,(\nabla\cdot\underline{\mathbf{U}})^t + \nabla p &=& \mathbf{f}, & \text{in } \Omega,\\ \nabla\cdot\mathbf{u} + \delta p &=& g, & \text{in } \Omega,\\ \nabla tr\,\underline{\mathbf{U}} + \delta\nabla p &=& \nabla g, & \text{in } \Omega,\\ \nabla\times\underline{\mathbf{U}} &=& \underline{\mathbf{0}}, & \text{in } \Omega,\\ \mathbf{u} &=& \mathbf{0}, & \text{on } \partial\Omega,\\ \mathbf{n}\times\underline{\mathbf{U}} &=& \underline{\mathbf{0}}, & \text{on } \partial\Omega. \end{array}\right.$$

Let $\mathcal{D}(\Omega)$ be the linear space of infinitely differentiable functions with compact support on Ω and let $\mathcal{D}'(\Omega)$ denote the dual space of $\mathcal{D}(\Omega)$. The duality pairing between $\mathcal{D}'(\Omega)$ and $\mathcal{D}(\Omega)$ is denoted by $<\cdot,\cdot>$. We use the standard notation and definition for the Sobolev spaces $H^s(\Omega)^n$ and $H^s(\partial\Omega)^n$ for $s \geq 0$; the standard associated inner products are denoted by $(\cdot,\cdot)_{s,\Omega}$ and $(\cdot,\cdot)_{s,\partial\Omega}$, and their respective norms by $\|\cdot\|_{s,\Omega}$ and $\|\cdot\|_{s,\partial\Omega}$. (We suppress

the superscript n because dependence of the vector norms on dimension will be clear by context. We also omit Ω from the inner product and norm designation when there is no risk of confusion.) For $s = 0$, $H^s(\Omega)^n$ coincides with $L^2(\Omega)^n$. In this case, the norm and inner product will be denoted by $\|\cdot\|$ and $(\cdot, \cdot)$, respectively. As usual, $H_0^s(\Omega)$ is the closure of $\mathcal{D}(\Omega)$ with respect to the norm $\|\cdot\|_s$, and $H^{-s}(\Omega)$ is its dual with norm defined by

$$\|\varphi\|_{-s} = \sup_{0 \neq \phi \in H_0^s(\Omega)} \frac{< \varphi, \phi >}{\|\phi\|_s}.$$

Define the product spaces $H_0^s(\Omega)^n = \prod_{i=1}^n H_0^s(\Omega)$ and $H^{-s}(\Omega)^n = \prod_{i=1}^n H^{-s}(\Omega)$ with standard product norms. Let

$$H(\mathrm{div};\, \Omega) = \{\mathbf{v} \in L^2(\Omega)^n \,:\, \nabla \cdot \mathbf{v} \in L^2(\Omega)\}$$

and

$$H(\mathbf{curl};\, \Omega) = \{\mathbf{v} \in L^2(\Omega)^n \,:\, \nabla \times \mathbf{v} \in L^2(\Omega)^{2n-3}\},$$

which are Hilbert spaces under the respective norms

$$\|\mathbf{v}\|_{H(\mathrm{div};\Omega)} \equiv \left(\|\mathbf{v}\|^2 + \|\nabla \cdot \mathbf{v}\|^2\right)^{\frac{1}{2}}$$

and

$$\|\mathbf{v}\|_{H(\mathbf{curl};\Omega)} \equiv \left(\|\mathbf{v}\|^2 + \|\nabla \times \mathbf{v}\|^2\right)^{\frac{1}{2}}.$$

Define their subspaces

$$H_0(\mathrm{div};\, \Omega) = \{\mathbf{v} \in H(\mathrm{div};\, \Omega) \,:\, \mathbf{n} \cdot \mathbf{v} = 0 \text{ on } \partial\Omega\}$$

and

$$H_0(\mathbf{curl};\, \Omega) = \{\mathbf{v} \in H(\mathbf{curl};\, \Omega) \,:\, \gamma_\tau \mathbf{v} = 0 \text{ on } \partial\Omega\},$$

where $\gamma_\tau \mathbf{v} = \boldsymbol{\tau} \cdot \mathbf{v}$ for $n = 2$ and $\gamma_\tau \mathbf{v} = \mathbf{n} \times \mathbf{v}$ for $n = 3$; $\mathbf{n}$ and $\boldsymbol{\tau}$ denote the respective unit vectors normal and tangent to the boundary. Finally, define

$$L_0^2(\Omega)^n = \{\mathbf{v} \in L^2(\Omega)^n \,:\, \int_\Omega v_i \, dx = 0 \text{ for } i = 1, ..., n\}.$$

It is well-known that the (weak form of the) boundary value problem (2.1)-(2.2) has a unique solution $(\mathbf{u}, p) \in H_0^1(\Omega)^n \times L_0^2(\Omega)$ for any $\mathbf{f} \in H^{-1}(\Omega)^n$ and for $g \in H^1(\Omega)$ (e.g., see [16, 17, 14]). Moreover, if the boundary of the domain Ω is $C^{1,1}$ or a convex polyhedron, then the following H^2-regularity result holds:

$$\|\nu\, \mathbf{u}\|_2 + \|p\|_1 \leq C\, (\|\mathbf{f}\|_0 + \|\nu\, g\|_1)\,. \tag{2.9}$$

(We use C with or without subscripts in this paper to denote a generic positive constant, possibly different at different occurrences, which is independent of the Reynolds number and other parameters introduced in this paper but may depend on the domain Ω or the constant δ.) Bound (2.9) is established for the case $\nu = 1$ and $\delta = 0$ in [16, 17]; the case for

general ν and $\delta = 0$ is then immediate. The case $\delta > 0$ follows from the well-known linear elasticity bound $\|\mathbf{u}\|_2 + \|\sigma\|_1 \leq C\,\|\mathbf{f}\|_0$, where $\mathbf{f}$ is the (unscaled) source term in (2.4) and σ is the stress tensor. We will need (2.9) to establish full H^1 product ellipticity of one of our reformulations of (2.1)-(2.2); see Theorem 3.2.

The following lemma is an immediate consequence of a general functional analysis result due to Nečas [18] (see also [14]).

LEMMA 2.1. *For any p in $L_0^2(\Omega)$, we have*

$$\|p\| \leq C\,\|\nabla p\|_{-1}. \tag{2.10}$$

Proof. See [18] for a general proof. □

A curl result analogous to Green's theorem for divergence follows from [14] (Theorem 2.11 in Chapter I):

$$(\nabla\times \mathbf{z},\ \boldsymbol{\phi}) = (\mathbf{z},\ \nabla\times\boldsymbol{\phi}) - \int_{\partial\Omega} \boldsymbol{\phi}\cdot(\mathbf{n}\times\mathbf{z})\,ds \tag{2.11}$$

for $\mathbf{z} \in H(\mathbf{curl};\,\Omega)$ and $\boldsymbol{\phi} \in H^1(\Omega)^n$.

Finally, we summarize results from [14] that we will need for G_2 in the next section. The first inequality follows from Theorems 3.7–3.9 in [14], while the second inequality follows from Lemmas 3.4 and 3.6 in [14].

THEOREM 2.1. *Assume that the domain Ω is a bounded convex polyhedron or has $C^{1,1}$ boundary. Then for any vector function $\mathbf{v}$ in either $H_0(\mathrm{div};\,\Omega)\cap H(\mathbf{curl};\,\Omega)$ or $H(\mathrm{div};\,\Omega)\cap H_0(\mathbf{curl};\,\Omega)$, we have*

$$\|\mathbf{v}\|_1^2 \leq C\left(\|\mathbf{v}\|^2 + \|\nabla\cdot\mathbf{v}\|^2 + \|\nabla\times\mathbf{v}\|^2\right). \tag{2.12}$$

If, in addition, the domain is simply connected, then

$$\|\mathbf{v}\|_1^2 \leq C\left(\|\nabla\cdot\mathbf{v}\|^2 + \|\nabla\times\mathbf{v}\|^2\right). \tag{2.13}$$

3. First-Order System Least Squares. In this section, we consider least-squares functionals based on system (2.6) and its extension (2.8). Our primary objective here is to establish ellipticity of these least-squares functionals in the appropriate Sobolev spaces.

Our first least-squares functional is defined in terms of appropriate weights and norms of the residuals for system (2.6):

$$\begin{aligned} G_1(\underline{\mathbf{U}},\,\mathbf{u},\,p;\,\mathbf{f},\,g) \;&=\; \|\mathbf{f} + \nu\,(\nabla\cdot\underline{\mathbf{U}})^t - \nabla p\|_{-1}^2 + \nu^2\,\|\underline{\mathbf{U}} - \nabla\,\mathbf{u}^t\|^2 \\ &\quad + \nu^2\,\|\nabla\cdot\mathbf{u} + \delta\,p - g\|^2. \end{aligned} \tag{3.1}$$

Note the use of the H^{-1} norm in the first term here. Our second functional is defined as a weighted sum of the L^2 norms of the residuals for system (2.8):

$$\begin{aligned} G_2(\underline{\mathbf{U}},\,\mathbf{u},\,p;\,\mathbf{f},\,g) \;&=\; \|\mathbf{f} + \nu\,(\nabla\cdot\underline{\mathbf{U}})^t - \nabla p\|^2 + \nu^2\,\|\underline{\mathbf{U}} - \nabla\,\mathbf{u}^t\|^2 \\ &+\; \nu^2\,\|\nabla\cdot\mathbf{u} + \delta\,p - g\|^2 + \nu^2\,\|\nabla\times\underline{\mathbf{U}}\|^2 + \nu^2\,\|\nabla\,tr\,\underline{\mathbf{U}} + \delta\nabla\,p - \nabla g\|^2. \end{aligned} \tag{3.2}$$

Let

$$\mathbf{V}_1 = L^2(\Omega)^{n^2}\times H_0^1(\Omega)^n \times L_0^2(\Omega) \quad\text{and}\quad \mathbf{V}_2 = \underline{\mathbf{V}}_0 \times H_0^1(\Omega)^n \times (H^1(\Omega)/\Re),$$

where

$$\underline{\mathbf{V}}_0 \equiv \{\underline{\mathbf{V}} \in H^1(\Omega)^{n^2} \ : \ \mathbf{n} \times \underline{\mathbf{V}} = \underline{\mathbf{0}} \quad on\ \partial\Omega\}.$$

Note that $\mathbf{V}_2 \subset \mathbf{V}_1$. For $i = 1$ or 2, the first-order system least-squares variational problem for the Stokes equations is to minimize the quadratic functional $G_i(\underline{\mathbf{U}}, \mathbf{u}, p; \mathbf{f}, g)$ over $\mathbf{V}_i$: find $(\underline{\mathbf{U}}, \mathbf{u}, p) \in \mathbf{V}_i$ such that

$$G_i(\underline{\mathbf{U}}, \mathbf{u}, p; \mathbf{f}, g) = \inf_{(\underline{\mathbf{V}}, \mathbf{v}, q) \in \mathbf{V}_i} G_i(\underline{\mathbf{V}}, \mathbf{v}, q; \mathbf{f}, g). \tag{3.3}$$

THEOREM 3.1. *There exists a constant C independent of ν such that for any $(\underline{\mathbf{U}}, \mathbf{u}, p) \in \mathbf{V}_1$ we have*

$$\frac{1}{C}\left(\nu^2\|\underline{\mathbf{U}}\|^2 + \nu^2\|\mathbf{u}\|_1^2 + \|p\|^2\right) \le G_1(\underline{\mathbf{U}}, \mathbf{u}, p; \mathbf{0}, 0) \tag{3.4}$$

and

$$G_1(\underline{\mathbf{U}}, \mathbf{u}, p; \mathbf{0}, 0) \le C\left(\nu^2\|\underline{\mathbf{U}}\|^2 + \nu^2\|\mathbf{u}\|_1^2 + \|p\|^2\right). \tag{3.5}$$

Proof. Upper bound (3.5) is straightforward from the triangle and Cauchy-Schwarz inequalities. We proceed to show the validity of (3.4) for $(\underline{\mathbf{U}}, \mathbf{u}, p) \in \mathbf{W}_1 \equiv \{H(\mathrm{div}; \Omega)^n \times H_0^1(\Omega)^n \times (L_0^2(\Omega) \cap H^1(\Omega))\}$. Then (3.4) would follow for $(\underline{\mathbf{U}}, \mathbf{u}, p) \in \mathbf{V}_1$ by continuity. For any $(\underline{\mathbf{U}}, \mathbf{u}, p) \in \mathbf{W}_1$ and $\boldsymbol{\phi} \in H_0^1(\Omega)^n$, we have

$$\begin{aligned}(\nabla p, \boldsymbol{\phi}) &= (-\nu(\nabla \cdot \underline{\mathbf{U}})^t + \nabla p, \boldsymbol{\phi}) - \nu(\underline{\mathbf{U}}, \nabla \boldsymbol{\phi}^t) \\ &\le \| - \nu(\nabla \cdot \underline{\mathbf{U}})^t + \nabla p\|_{-1}\|\boldsymbol{\phi}\|_1 + \nu\|\underline{\mathbf{U}}\|\,\|\nabla \boldsymbol{\phi}^t\|.\end{aligned}$$

Hence, by Lemma 2.1, we have

$$\|p\| \le C\left(\| - \nu(\nabla \cdot \underline{\mathbf{U}})^t + \nabla p\|_{-1} + \nu\|\underline{\mathbf{U}}\|\right). \tag{3.6}$$

From (3.6) and the Poincaré-Friedrichs inequality on $\mathbf{u}$ we have

$$\begin{aligned}&\nu^2\|\nabla \mathbf{u}^t\|^2 \\ =\ & \nu^2(\nabla \mathbf{u}^t - \underline{\mathbf{U}}, \nabla \mathbf{u}^t) + \nu(-\nu(\nabla \cdot \underline{\mathbf{U}})^t + \nabla p, \mathbf{u}) + \nu(p, \nabla \cdot \mathbf{u} + \delta p) - \nu\delta(p,p) \\ \le\ & \nu^2\|\nabla \mathbf{u}^t - \underline{\mathbf{U}}\|\,\|\nabla \mathbf{u}^t\| + \nu\| - \nu(\nabla \cdot \underline{\mathbf{U}})^t + \nabla p\|_{-1}\|\mathbf{u}\|_1 + \nu\|p\|\,\|\nabla \cdot \mathbf{u} + \delta p\| \\ \le\ & \left(\nu\|\nabla \mathbf{u}^t - \underline{\mathbf{U}}\| + C\| - \nu(\nabla \cdot \underline{\mathbf{U}})^t + \nabla p\|_{-1}\right)\nu\|\nabla \mathbf{u}^t\| \\ & + C\| - \nu(\nabla \cdot \underline{\mathbf{U}})^t + \nabla p\|_{-1}\nu\|\nabla \cdot \mathbf{u} + \delta p\| + C\nu^2\|\underline{\mathbf{U}}\|\,\|\nabla \cdot \mathbf{u} + \delta p\|.\end{aligned}$$

Using the ε-inequality, $2ab \le \frac{1}{\varepsilon}a^2 + \varepsilon b^2$, with $\varepsilon = 1$ for the first two products yields

$$\nu^2\|\nabla \mathbf{u}^t\|^2 \le CG_1(\underline{\mathbf{U}}, \mathbf{u}, p; \mathbf{0}, 0) + C\nu^2\|\underline{\mathbf{U}}\|\,\|\nabla \cdot \mathbf{u} + \delta p\|. \tag{3.7}$$

Again from (3.6) and the Poincaré-Friedrichs inequality on $\mathbf{u}$ we have

$$\begin{aligned}&\nu^2\|\underline{\mathbf{U}}\|^2 \\ =\ & \nu^2(\underline{\mathbf{U}} - \nabla \mathbf{u}^t, \underline{\mathbf{U}}) + \nu(\mathbf{u}, -\nu(\nabla \cdot \underline{\mathbf{U}})^t + \nabla p) + \nu(\nabla \cdot \mathbf{u} + \delta p, p) - \nu\delta(p,p) \\ \le\ & \nu^2\|\underline{\mathbf{U}} - \nabla \mathbf{u}^t\|\,\|\underline{\mathbf{U}}\| + C\nu\|\nabla \mathbf{u}^t\|\,\| - \nu(\nabla \cdot \underline{\mathbf{U}})^t + \nabla p\|_{-1} + \nu\|p\|\,\|\nabla \cdot \mathbf{u} + \delta p\| \\ \le\ & \nu^2\|\underline{\mathbf{U}} - \nabla \mathbf{u}^t\|\,\|\underline{\mathbf{U}}\| + C\nu\|\nabla \mathbf{u}^t\|\,\| - \nu(\nabla \cdot \underline{\mathbf{U}})^t + \nabla p\|_{-1} \\ & + C\| - \nu(\nabla \cdot \underline{\mathbf{U}})^t + \nabla p\|_{-1}\nu\|\nabla \cdot \mathbf{u} + \delta p\| + C\nu^2\|\underline{\mathbf{U}}\|\,\|\nabla \cdot \mathbf{u} + \delta p\|.\end{aligned}$$

Using the ε-inequality on the first three products and (3.7), we then have

$$\begin{aligned}\nu^2\|\underline{\mathbf{U}}\|^2 &\leq CG_1(\underline{\mathbf{U}}, \mathbf{u}, p; \mathbf{0}, 0) + C\nu^2\|\nabla \mathbf{u}^t\|^2 + C\nu^2\|\underline{\mathbf{U}}\|\,\|\nabla\cdot\mathbf{u} + \delta p\| \\ &\leq CG_1(\underline{\mathbf{U}}, \mathbf{u}, p; \mathbf{0}, 0) + C\nu^2\|\underline{\mathbf{U}}\|\,\|\nabla\cdot\mathbf{u} + \delta p\|.\end{aligned}$$

Again using the ε-inequality, we find that

$$\nu^2\|\underline{\mathbf{U}}\|^2 \leq CG_1(\underline{\mathbf{U}}, \mathbf{u}, p; \mathbf{0}, 0). \tag{3.8}$$

Using (3.8) in (3.6) and (3.7), we now have that

$$\|p\|^2 \leq CG_1(\underline{\mathbf{U}}, \mathbf{u}, p; \mathbf{0}, 0) \quad \text{and} \quad \nu^2\|\nabla \mathbf{u}^t\|^2 \leq CG_1(\underline{\mathbf{U}}, \mathbf{u}, p; \mathbf{0}, 0).$$

The theorem now follows from these bounds, (3.8), and the Poincaré-Friedrichs inequality on $\mathbf{u}$. □

The next two lemmas will be useful in the proof of Theorem 3.2.

LEMMA 3.1. *(Poincaré-Friedrichs-type inequality.) Suppose that the assumptions of* Theorem 2.1 *hold. Let $p \in H^1(\Omega)$ satisfy $\int_\Omega p\,dz = 0$; then*

$$\|p\| \leq C|p|_1, \tag{3.9}$$

where C depends only on Ω. Further, let $\mathbf{q} \in (H_0^1(\Omega) \cap H^2(\Omega))^n$; then

$$\|\nabla\cdot\mathbf{q}\| \leq C|\nabla\cdot\mathbf{q}|_1, \tag{3.10}$$

where C depends only on Ω.

Proof. Equation $\int_\Omega p\,dz = 0$ implies $p = 0$ at some point in Ω. The first result now follows from the standard Poincaré-Friedrichs inequality. The second result follows from the fact that $\int_\Omega \nabla\cdot\mathbf{q}\,dz = 0$. □

LEMMA 3.2. *Under the assumptions of* Theorem 2.1 *with simply connected Ω, for any p in $H^1(\Omega)$ we have:*

($n = 2$) let $\boldsymbol{\phi} = (\phi_1, \phi_2)^t$ and $\mathbf{q} = (q_1, q_2)^t$; if each $q_i \in H_0^1(\Omega)\cap H^2(\Omega)$ and each $\phi_i \in H^1(\Omega)$ is such that $\Delta\phi_i \in L^2(\Omega)$ and $\mathbf{n}\cdot\nabla\phi_i = 0$ on $\partial\Omega$, then

$$|\nabla\cdot\mathbf{q} + \delta p|_1^2 \leq C\left(|\nabla\cdot\mathbf{q} + tr\,\nabla^\perp\boldsymbol{\phi}^t + \delta p|_1^2 + \|\Delta\boldsymbol{\phi}\|^2\right); \tag{3.11}$$

($n = 3$) let $\underline{\Phi} = (\phi_1, \phi_2, \phi_3)$ and $\mathbf{q} = (q_1, q_2, q_3)^t$; if each $q_i \in H_0^1(\Omega) \cap H^2(\Omega)$ and each $\boldsymbol{\phi}_i \in H^1(\Omega)^3$ is divergence free with $\Delta\boldsymbol{\phi}_i \in L^2(\Omega)^n$ and $\mathbf{n}\times(\nabla\times\boldsymbol{\phi}_i) = \mathbf{0}$ on $\partial\Omega$, then

$$|\nabla\cdot\mathbf{q} + \delta p|_1^2 \leq C\left(|\nabla\cdot\mathbf{q} + tr\,\nabla\times\underline{\Phi} + \delta p|_1^2 + \|\Delta\underline{\Phi}\|^2\right). \tag{3.12}$$

Proof. ($n = 2$) The assumptions of Theorem 2.1 are sufficient to guarantee H^2-regularity of the Laplace equation on Ω; that is, the second inequality in the equation

$$|\nabla\times\boldsymbol{\phi}|_1 \leq C\,|\boldsymbol{\phi}|_2 \leq C\,\|\Delta\boldsymbol{\phi}\|.$$

Note that $tr\,(\nabla^\perp\phi_1, \nabla^\perp\phi_2) = \nabla\times\boldsymbol{\phi}$. Then, from the above and the triangle inequality, we have

$$|\nabla\cdot\mathbf{q}+\delta p|_1^2 \leq 2\left(|\nabla\cdot\mathbf{q} + \nabla\times\boldsymbol{\phi} + \delta p|_1^2 + |\nabla\times\boldsymbol{\phi}|_1^2\right) \leq C\left(|\nabla\cdot\mathbf{q} + tr\,\nabla^\perp\boldsymbol{\phi}^t + \delta p|_1^2 + \|\Delta\boldsymbol{\phi}\|^2\right),$$

which is (3.11).
($n = 3$) Bound (2.13) with $\mathbf{v} = \nabla\times\underline{\Phi}$ and identity (2.5) applied to each column of $\nabla\times\underline{\Phi}$ imply that

$$|tr\,\nabla\times\underline{\Phi}|_1^2 \le 3\,|\nabla\times\underline{\Phi}|_1^2 \le C\left(\|\nabla\cdot\nabla\times\underline{\Phi}\|^2 + \|\nabla\times\nabla\times\underline{\Phi}\|^2\right) = C\,\|\Delta\underline{\Phi}\|^2$$

since each ϕ_i is divergence free. Eqn. (3.12) now follows from the triangle inequality as for the case $n = 2$. □

THEOREM 3.2. *Assume that the domain Ω is a bounded convex polyhedron or has $C^{1,1}$ boundary and that regularity bound (2.9) holds. Then, there exists a constant C independent of ν such that for any $(\underline{\mathbf{U}}, \mathbf{u}, p) \in \mathbf{V}_2$, we have*

$$\frac{1}{C}\left(\nu^2\,\|\underline{\mathbf{U}}\|_1^2 + \nu^2\,\|\mathbf{u}\|_1^2 + \|p\|_1^2\right) \le G_2(\underline{\mathbf{U}}, \mathbf{u}, p; \mathbf{0}, 0) \tag{3.13}$$

and

$$G_2(\underline{\mathbf{U}}, \mathbf{u}, p; \mathbf{0}, 0) \le C\left(\nu^2\,\|\underline{\mathbf{U}}\|_1^2 + \nu^2\,\|\mathbf{u}\|_1^2 + \|p\|_1^2\right). \tag{3.14}$$

Proof. Upper bound (3.14) is straightforward from the triangle and Cauchy-Schwarz inequalities. To prove (3.13), note that the H^{-1} norm of a function is always bounded by its L^2 norm. Since $\mathbf{V}_2 \subset \mathbf{V}_1$, then $G_1 \le G_2$ on $\mathbf{V}_2$. Hence, by Theorem 3.1, we have

$$\nu^2\|\underline{\mathbf{U}}\|^2 + \nu^2\,\|\mathbf{u}\|_1^2 + \|p\|^2 \le CG_1(\underline{\mathbf{U}}, \mathbf{u}, p; \mathbf{0}, 0) \le CG_2(\underline{\mathbf{U}}, \mathbf{u}, p; \mathbf{0}, 0). \tag{3.15}$$

From Theorem 2.1 and (3.9), we have

$$\nu^2\,\|\underline{\mathbf{U}}\|_1^2 + \|p\|_1^2 \le C\left(\nu^2\,\|\underline{\mathbf{U}}\|^2 + \nu^2\,\|(\nabla\cdot\underline{\mathbf{U}})^t\|^2 + \nu^2\,\|\nabla\times\underline{\mathbf{U}}\|^2 + \|\nabla\,p\|^2\right). \tag{3.16}$$

It thus suffices to show that

$$\begin{aligned} &C\left(\nu^2\,\|(\nabla\cdot\underline{\mathbf{U}})^t\|^2 + \|\nabla\,p\|^2\right) \\ &\qquad \le \|-\nu(\nabla\cdot\underline{\mathbf{U}})^t + \nabla\,p\|^2 + \nu^2\,|tr\,\underline{\mathbf{U}} + \delta p|_1^2 + \nu^2\,\|\nabla\times\underline{\mathbf{U}}\|^2. \end{aligned} \tag{3.17}$$

We will prove (3.17) only for the case $n = 3$ because the proof for $n = 2$ is similar. First, we assume that the domain Ω is simply connected with connected boundary. Since $\mathbf{n}\times\underline{\mathbf{U}} = \underline{\mathbf{0}}$ on $\partial\Omega$, the following decomposition is admitted :

$$\underline{\mathbf{U}} = \nabla\,\mathbf{q}^t + \nabla\times\underline{\Phi}, \tag{3.18}$$

where $\mathbf{q} \in H_0^1(\Omega)^n \cap H^2(\Omega)^n$ and $\underline{\Phi}$ is columnwise divergence free with $\mathbf{n}\times(\nabla\times\underline{\Phi}) = \underline{\mathbf{0}}$ on $\partial\Omega$. Here, we choose $\mathbf{q}$ to satisfy

$$\left\{\begin{array}{rcll} \Delta\mathbf{q} &=& (\nabla\cdot\underline{\mathbf{U}})^t, & \text{in } \Omega, \\ \mathbf{q} &=& \mathbf{0}, & \text{on } \delta\Omega, \end{array}\right. \tag{3.19}$$

Then, $\underline{\mathbf{V}} = \underline{\mathbf{U}} - \nabla\mathbf{q}^t$ is divergence free and satisfies $\mathbf{n}\times\underline{\mathbf{V}} = \mathbf{0}^t$. Since Ω has connected boundary we know that $\int_\Gamma \mathbf{n}\cdot\underline{\mathbf{V}} = \mathbf{0}^t$. Thus, Theorem 3.4 in [14] yields $\underline{\mathbf{V}} = \nabla\times\underline{\Phi}$, where $\nabla\cdot\underline{\Phi} = \mathbf{0}^t$.

By taking the curl of both sides of this decomposition, it is easy to see that

$$\|\Delta \underline{\Phi}\| = \|\nabla \times \underline{U}\| \leq \|\underline{U}\|_1, \tag{3.20}$$

so that $\|\Delta \underline{\Phi}\|$ is bounded and Lemma 3.2 applies. Hence,

$$\begin{aligned}
\| -\nu(\nabla \cdot \underline{U})^t \;+\; & \nabla p\|^2 + \nu^2 |tr\, \underline{U} + \delta p|_1^2 + \nu^2 \|\nabla \times \underline{U}\|^2 \\
& \text{(by equation (3.18))} \\
= \; & \| -\nu \Delta \mathbf{q} + \nabla p\|^2 + \nu^2 |\nabla \cdot \mathbf{q} + tr\, \nabla \times \underline{\Phi} + \delta p|_1^2 + \nu^2 \|\Delta \underline{\Phi}\|^2 \\
& \text{(by Lemma 3.2)} \\
\geq \; & \| -\nu \Delta \mathbf{q} + \nabla p\|^2 + C\nu^2 |\nabla \cdot \mathbf{q} + \delta p|_1^2 \\
& \text{(by Lemma 3.1)} \\
\geq \; & \| -\nu \Delta \mathbf{q} + \nabla p\|^2 + C\nu^2 \|\nabla \cdot \mathbf{q} + \delta p\|_1^2 \\
& \text{(by regularity assumption (2.9) with } \mathbf{u} = \mathbf{q}) \\
\geq \; & C\left(\nu^2 \|\Delta \mathbf{q}\|^2 + \|\nabla p\|^2\right) \\
& \text{(by equation (3.18))} \\
= \; & C\left(\nu^2 \|\nabla \cdot \underline{U}\|^2 + \|\nabla p\|^2\right).
\end{aligned}$$

This proves (3.17) and, hence, the theorem for simply connected Ω.

The proof for general Ω (i.e., when we assume only that $\partial\Omega$ is $C^{1,1}$) now follows by an argument similar to the proof of Theorem 3.7 in [14]. □

We now show that the last two terms in the definition of G_2 are necessary for the bound (3.13) to hold, even with the extra boundary condition $\mathbf{n} \times \underline{U} = \underline{0}$. We consider the Stokes equations, so that $\delta = 0$. Suppose first that we omit the term $\|\nabla \times \underline{U}\|^2$ but include the term $\|\nabla tr\, \underline{U}\|^2$. We offer a two-dimensional counterexample; a three-dimensional counterexample can be constructed in a similar manner. Let $\nu = 1$, $\mathbf{u} = \mathbf{0}$, and $p = 0$. Choose any $\omega \in \mathcal{D}(\Omega)$ such that $\Delta \nabla \omega \neq \mathbf{0}$ and define

$$\underline{U} \equiv \nabla^{\perp}(\nabla \omega)^t.$$

Clearly, $\mathbf{n} \times \underline{U} = \underline{0}$. It is easy to show that

$$\nabla \cdot \underline{U} = \mathbf{0} \quad \text{and} \quad tr\, \underline{U} = \nabla \times (\nabla \omega) = 0.$$

However,

$$(\nabla \times \underline{U})^t = \Delta \nabla \omega \neq \mathbf{0}$$

by construction. Thus,

$$G_2(\underline{U}, \mathbf{u}, p; \mathbf{0}) = \|\underline{U}\|^2,$$

which cannot bound $\|\underline{U}\|_1^2$. That is, since $\omega \in \mathcal{D}(\Omega)$ is arbitrary, we may choose it so oscillatory that $\|\underline{U}\|_1 / \|\underline{U}\|$ is as large as we like. This prevents the bound (3.13) from holding.

Next suppose we include the $\|\nabla \times \underline{U}\|^2$ term but omit the $\|\nabla tr\, \underline{U}\|^2$ term. Now set $\Omega = (0, 1)^2$, $\nu = 1$, $\mathbf{u} = \mathbf{0}$, and $p = \cos(k\pi x_1)\sin(\pi x_2)$ and choose q_i to satisfy

$$\begin{cases} -\Delta q_i = -\partial_i p, & \text{in} \quad \Omega, \\ q_i = 0, & \text{on} \quad \partial\Omega, \end{cases}$$

for $i = 1, 2$. Then

$$q_1 = \frac{k}{\pi(k^2+1)} \sin(k\pi x_1) \sin(\pi x_2).$$

We also know that

$$\|\nabla q_2\| \leq C \|\partial_2 p\| = \|\pi \cos(k\pi x_1) \cos(\pi x_2)\| \leq C,$$

where C is independent of k. Now set

$$\mathbf{U}_i = \nabla q_i$$

for $i = 1, 2$. Then $\mathbf{n} \times \mathbf{U}_i = \mathbf{0}$ and

$$G_2(\underline{\mathbf{U}}, \mathbf{u}, p; \mathbf{0}) = \|\Delta \mathbf{q} - \nabla p\|^2 + \|\nabla \mathbf{q}\|^2 = \|\nabla \mathbf{q}\|^2 \leq C,$$

where C is independent of k. On the other hand, we have

$$\|p\|_1 \geq C k,$$

which again prevents the bound (3.13) from holding.

4. Concluding Remarks. Full regularity assumption (2.9) is needed in Theorem 3.2 only to obtain full H^1 product ellipticity of augmented functional G_2 in (3.2). This somewhat restrictive assumption is not necessary for functional G_1 in (3.1), which supports an efficient practical algorithm (the H^{-1} norm in (3.1) can be replaced by a discrete inverse norm or a simpler mesh weighted norm; see [5] and [8] for analogous inverse norm algorithms) and which has the weaker norm equivalence assured by Theorem 3.1.

Nevertheless, the principal result of this paper is Theorem 3.2, which establishes full H^1 product ellipticity of least-squares functional G_2 for the generalized Stokes system. Since we have assumed full H^2-regularity of the original Stokes (linear elasticity) equations, we may then use this result to establish optimal finite element approximation estimates and optimal multiplicative and additive multigrid convergence rates. This can be done in precisely the same way that these results were established for general second-order elliptic equations (see [10], Sections 3-5). We therefore omit this development here. However, it is important to recognize that the ellipticity property is independent of the Reynolds parameter ν (Lamé constants μ and λ). This automatically implies that the optimal finite element discretization error estimates and multigrid convergence factor bounds are uniform in ν (λ and μ). At first glance, it might appear that the scaling of some of the H^1 product norm components might create a scale dependence of our discretization and algebraic convergence estimates. However, the results in [10] are based only on assumptions posed in an unscaled H^1 product norm, in which the individual variables are completely decoupled; and since the constant ν appears only as a simple factor in individual terms of the scaled H^1 norm, these assumptions are equally valid in this case. On the other hand, for problems where the necessary H^1 scaling is not (essentially) constant, extension of the theory of Sections 3–5 of [10] is not straightforward. Such is the case for convection-diffusion equations, which will be treated in a forthcoming paper.

REFERENCES

[1] S. AGMON, A. DOUGLIS, AND L. NIRENBERG, *Estimates near the boundary for solutions of elliptic partial differential equations satisfying general boundary conditions II*, Com. on Pure Appl. Math., Vol. 17, (1964), pp. 35-92.

[2] A. K. AZIZ, R. B. KELLOGG, AND A. B. STEPHENS, *Least-squares methods for elliptic systems*, Math. Comp., 44 (1985), pp. 53–70.

[3] P. B. BOCHEV, *Personal communication*, San Diego, July, 1994.

[4] P. B. BOCHEV, *Analysis of least-squares finite element methods for the Navier-Stokes equations*, submitted.

[5] P. B. BOCHEV AND M. D. GUNZBURGER, *Accuracy of least-squares methods for the Navier–Stokes equations*, Comput. Fluids, 22 (1993), pp. 549–563.

[6] P. B. BOCHEV AND M. D. GUNZBURGER, *Analysis of least-squares finite element methods for the Stokes equations*, Math. Comp., to appear.

[7] P. B. BOCHEV AND M. D. GUNZBURGER, *Least-squares methods for the velocity-pressure-stress formulation of the Stokes equations*, Comput. Methods Appl. Mech. Engrg., to appear.

[8] J. H. BRAMBLE, R. D. LAZAROV, AND J. E. PASCIAK, *A least-squares approach based on a discrete minus one inner product for first order system*, manuscript.

[9] Z. CAI, R. D. LAZAROV, T. MANTEUFFEL, AND S. MCCORMICK, *First-order system least squares for partial differential equations: Part I*, SIAM J. Numer. Anal., 31 (1994).

[10] Z. CAI, T. MANTEUFFEL, AND S. MCCORMICK, *First-order system least squares for partial differential equations: Part II*, SIAM J. Numer. Anal, submitted.

[11] C. L. CHANG , *A mixed finite element method for the Stokes problem: an acceleration-pressure formulation* Appl. Math. Comp., 36 (1990), pp. 135-146.

[12] C. L. CHANG , *An error estimate of the least squares finite element methods for the Stokes problem in three dimensions* Math. Comp., 63 (1994), pp. 41-50.

[13] L. P. FRANCA AND R. STENBERG, *Error analysis of some Galerkin least squares methods for the linear elasticity equations*, SIAM J. Numer. Anal., 28 (1991), pp. 1680-1697.

[14] V. GIRAULT AND P. A. RAVIART, *Finite Element Methods for Navier-Stokes Equations: Theory and Algorithms*, Springer-Verlag, New York, 1986.

[15] B. JIANG, C. LOH, AND L. POVINELLI, *Theoretical study of the incompressible Navier-Stokes equations by the least-squares method*, NASA Tech. Memo. 106535, ICOMP-94-04.

[16] R. B. KELLOGG AND J. E. OSBORN, *A regularity result for the Stokes problem in a convex polygon*, J. Funct. Anal., 21 (1976), pp. 397–431.

[17] O. A. LADYZHENSKAYA, *The Mathematical Theory of Viscous Incompressible Flow*, Gordon and Breach, New York, 1963.

[18] J. NEČAS, *Equations aux Dérivées Partielles*, Presses de l'Université de Montréal, 1965.

[19] R. TÉMAM, *Navier-Stokes Equations*, North-Holland, New York, 1977.

Page intentionally left blank

TOWARDS AN FVE-FAC METHOD FOR DETERMINING THERMOCAPILLARY EFFECTS ON WELD POOL SHAPE

David Canright and Van Emden Henson
Mathematics Dept., Code MA
Naval Postgraduate School
Monterey, CA 93943

SUMMARY

Several practical materials processes, e.g., welding, float-zone purification, and Czochralski crystal growth, involve a pool of molten metal with a free surface, with strong temperature gradients along the surface. In some cases, the resulting thermocapillary flow is vigorous enough to convect heat toward the edges of the pool, increasing the driving force in a sort of positive feedback. In this work we examine this mechanism and its effect on the solid-liquid interface through a model problem: a half space of pure substance with concentrated axisymmetric surface heating, where surface tension is strong enough to keep the liquid free surface flat. The numerical method proposed for this problem utilizes a finite volume element (FVE) discretization in cylindrical coordinates. Because of the axisymmetric nature of the model problem, the control volumes used are torroidal prisms, formed by taking a polygonal cross-section in the (r, z) plane and sweeping it completely around the z-axis. Conservation of energy (in the solid), and conservation of energy, momentum, and mass (in the liquid) are enforced globally by integrating these quantities and enforcing conservation over each control volume. Judicious application of the Divergence Theorem and Stokes' Theorem, combined with a Crank-Nicolson time-stepping scheme leads to an implicit algebraic system to be solved at each time step.

It is known that near the boundary of the pool, that is, near the solid-liquid interface, the full conduction-convection solution will require extremely fine length scales to resolve the physical behavior of the system. Furthermore, this boundary moves as a function of time. Accordingly, we develop the foundation of an adaptive refinement scheme based on the principles of Fast Adaptive Composite Grid methods (FAC). Implementation of the method and numerical results will appear in a later report.

INTRODUCTION

Several practical materials processes, e.g., welding, float-zone purification, and Czochralski crystal growth, involve a pool of molten metal with a free surface, with

strong temperature gradients along the surface. In many cases (e.g., laser welding) convection in the liquid metal is driven primarily by thermocapillary forces, and even in cases where other forces are stronger overall, thermocapillary forces may still be dominant near the edge of the pool [4]. Previous work [2] showed how vigorous thermocapillary convection can lead to localized intense heat transfer and high velocities in the "cold corner" region where the liquid free surface meets the solid.

The present work examines how this localized heat transfer modifies the shape of the solid-liquid interface bounding the pool. When convection is vigorous, the high heat flux in the corner may melt away the solid near the surface, resulting in a sort of "lip" around the edge of the pool. This phenomenon is modeled computationally, and the steady solution sought for a wide range of the two governing parameters. This is a work in progress, in which numerical methods are proposed and developed for the problem. Implementation of the method and numerical results will appear in a later report.

PROBLEM STATEMENT

A half-space of a pure material is subjected to concentrated heating on the flat horizontal surface, giving a pool of molten material surrounded by solid. The total heat flux Q is constant, and far away the solid approaches the uniform cold temperature T_c (see Figure 1). Above the horizontal free surface is an inviscid, nonconducting gas. Surface tension of the liquid is assumed strong enough to keep the free surface flat (small Capillary number), but with surface tension variations due to a linear dependence on temperature. The resulting thermal and flow fields are assumed to be axisymmetric and steady, but the time-dependent equations are given below, to facilitate a numerical approach using time-like iterations to reach the steady solution.

Then the system is governed by conservation of energy in the solid and by conservation of energy, momentum, and mass in the pool:

$$\text{solid} \quad : \quad \frac{\partial T}{\partial t} = \kappa \nabla^2 T \tag{1}$$

$$\text{liquid} \quad : \quad \frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T = \kappa \nabla^2 T \tag{2}$$

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{u} \tag{3}$$

$$\nabla \cdot \mathbf{u} = 0 \tag{4}$$

with the conditions at the boundaries and at the solid-liquid interface given by

$$\text{solid surface} \quad (z = 0) \quad : \quad \frac{\partial T}{\partial z} = 0 \tag{5}$$

$$\text{liquid surface} \quad (z = 0) \quad : \quad k \frac{\partial T}{\partial z} = -q(r) \tag{6}$$

$$v = 0 \tag{7}$$

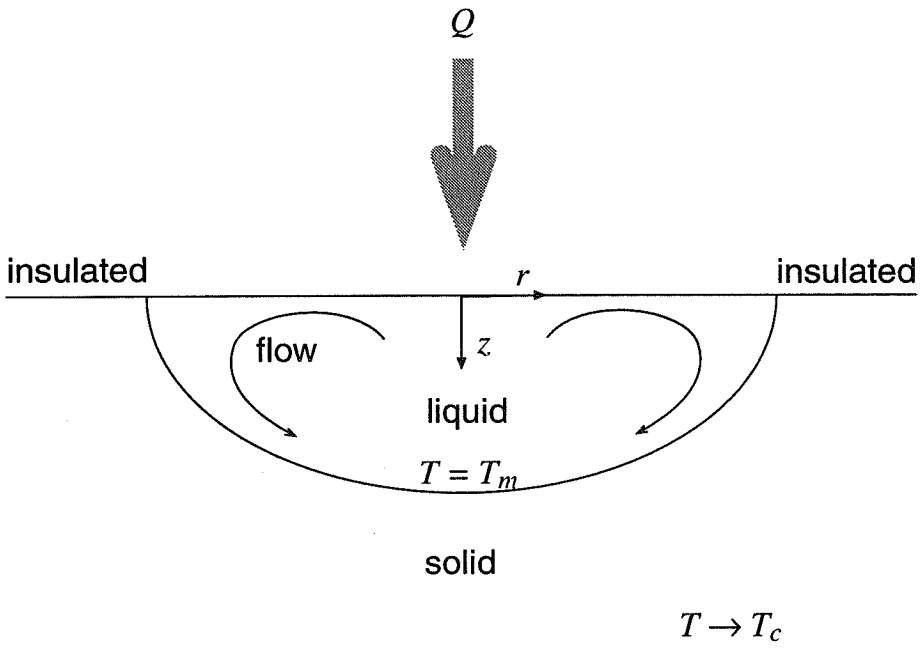


Figure 1: *Problem Formulation: a half-space of pure material is subjected to concentrated surface heating Q that results in a molten pool. (Outside the surface heating, the surface is adiabatic.) The melting temperature is T_m, and far away the solid is at the cooler temperature T_c. The flat liquid surface is subject to thermocapillary forcing, which drives convection in the liquid. Axisymmetry is assumed.*

$$\mu \frac{\partial u}{\partial z} = -\gamma \frac{\partial T}{\partial r} \tag{8}$$

$$\text{axis } (r = 0) \; : \quad \frac{\partial T}{\partial r} = 0 \tag{9}$$

$$u = 0 \tag{10}$$

$$\frac{\partial v}{\partial r} = 0 \tag{11}$$

$$\text{far away } (r, z \to \infty) \; : \quad T \to T_c \tag{12}$$

$$\text{interface } (r = f(z,t)) \; : \quad T = T_m \tag{13}$$

$$u = v = 0 \tag{14}$$

$$-(k \nabla T)_l = -(k \nabla T)_s + \rho L \mathbf{V}(z,t) \tag{15}$$

Here T is temperature, t is time, κ is thermal diffusivity, $\mathbf{u}$ is the velocity vector with components u and v in the r and z directions (cylindrical coordinates), ρ is density, p is pressure, ν is kinematic viscosity, k is thermal conductivity, $q(r)$ is the imposed surface heat flux (large at $r = 0$, falling off to zero at some small value of r, such that $\int_0^\infty q(r) 2\pi r \, dr = Q$), μ is viscosity, γ (assumed constant and positive) is the negative of the derivative of the surface tension with respect to temperature, T_m is the melting temperature, $r = f(z,t)$ gives the position of the solid-liquid interface, L is the latent heat of fusion, and $\mathbf{V}(z,t)$ is the normal velocity of the phase-change interface (that

is,

$$\mathbf{V}(z,t) = \frac{\partial f}{\partial t}/\sqrt{1+(\frac{\partial f}{\partial r})^2}\,\hat{\mathbf{n}}$$

where the unit normal vector is

$$\hat{\mathbf{n}} = (\hat{\mathbf{r}} + \frac{\partial f}{\partial r}\hat{\mathbf{z}})/\sqrt{1+(\frac{\partial f}{\partial r})^2}$$

in terms of the coordinate unit vectors).

To nondimensionalize the equations, we use a heat flux scale of Q and a temperature scale (relative to the cold temperature) of $\Delta T \equiv T_m - T_c$. Then thermal conduction gives the length scale $d \equiv Q/k\Delta T$ (so q scales as $Q/d^2 = (k\Delta T)^2/Q$), the thermocapillary coupling gives the velocity scale $u_s \equiv \gamma\,\Delta T/\mu$, and the convection time scale is $t_c \equiv d/u_s = \mu Q/k\gamma\Delta T^2$. The viscous pressure scale is $\mu u_s/d = k\gamma\Delta T^2/Q$. From the phase-change condition, the phase-change time scale is $t_p \equiv \rho L Q^2/(k\Delta T)^3$.

The resulting dimensionless equations are

$$\text{solid} \quad : \quad Ma\,\frac{\partial T}{\partial t} = \nabla^2 T \tag{16}$$

$$\text{liquid} \quad : \quad Ma\left(\frac{\partial T}{\partial t} + \mathbf{u}\cdot\nabla T\right) = \nabla^2 T \tag{17}$$

$$Re\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\,\mathbf{u}\right) = -\nabla p + \nabla^2\,\mathbf{u} \tag{18}$$

$$\nabla\cdot\mathbf{u} = 0 \tag{19}$$

with the boundary conditions

$$\text{solid surface} \ (z=0) \quad : \quad \frac{\partial T}{\partial z} = 0 \tag{20}$$

$$\text{liquid surface} \ (z=0) \quad : \quad \frac{\partial T}{\partial z} = -q(r) \tag{21}$$

$$v = 0 \tag{22}$$

$$\frac{\partial u}{\partial z} = \frac{\partial T}{\partial r} \tag{23}$$

$$\text{axis} \ (r=0) \quad : \quad \frac{\partial T}{\partial r} = 0 \tag{24}$$

$$u = 0 \tag{25}$$

$$\frac{\partial v}{\partial r} = 0 \tag{26}$$

$$\text{far away} \ (r, z \to \infty) \quad : \quad T \to 0 \tag{27}$$

$$\text{interface} \ (r = f(z,t)) \quad : \quad T = 1 \tag{28}$$

$$u = v = 0 \tag{29}$$

$$-\nabla T_l = -\nabla T_s + \lambda\mathbf{V} \tag{30}$$

where from this point on the variables denote the dimensionless quantities. The main dimensionless parameters are the Marangoni number $Ma \equiv u_s d/\kappa = \gamma Q/\mu k\kappa$ and

the Reynolds number $Re \equiv u_s d/\nu$. Their ratio gives the Prandtl number: $Pr \equiv \nu/\kappa = Ma/Re$. The other dimensionless parameter is the ratio of time scales, $\lambda \equiv t_p/t_c = \gamma QL/\nu k^2 \Delta T$, and so plays no role in the steady-state solution where $\mathbf{V} \to \mathbf{0}$.

For the numerical solutions, it is convenient to eliminate the pressure by adopting a stream-function/vorticity formulation for the flow:

$$Re\left(\frac{\partial \omega}{\partial t} - \nabla \times (\mathbf{u} \times \omega)\right) = -\nabla \times \nabla \times \omega \tag{31}$$

$$\omega = \nabla \times \nabla \times \left(\frac{\Psi}{r}\hat{\theta}\right) \tag{32}$$

$$u = -\frac{1}{r}\frac{\partial \Psi}{\partial z} \quad , \quad v = \frac{1}{r}\frac{\partial \Psi}{\partial r} \tag{33}$$

where Ψ is the axisymmetric stream function and ω is the vorticity vector (having only one component, in the $\hat{\theta}$ direction), with the flow boundary conditions

$$\text{liquid surface } (z = 0) \; : \; \Psi = 0 \tag{34}$$

$$\omega = \frac{\partial T}{\partial r} \tag{35}$$

$$\text{axis } (r = 0) \; : \; \Psi = 0 \tag{36}$$

$$\omega = \mathbf{0} \tag{37}$$

$$\text{interface } (r = f(z,t)) \; : \; \Psi = \frac{\partial \Psi}{\partial r} = \frac{\partial \Psi}{\partial z} = 0 \tag{38}$$

With the assumption of small Capillary number, the resulting small surface deflection can be determined as a small perturbation to the flat interface from the dimensionless normal stress condition at the surface:

$$-p + 2\frac{\partial v}{\partial z} = Ca^{-1}\frac{1}{r}\frac{d}{dr}\left(r\frac{dh}{dr}\right) \tag{39}$$

where $Ca \equiv \gamma \Delta T/\sigma$ is the Capillary number for surface tension σ, and the deflection $z = h(r)$ is taken positive upward. The contact line at the edge of the pool is assumed pinned ($h = 0$), and volume is conserved globally to determine the constant reference pressure level.

CONDUCTION SOLUTIONS

As a starting point for the numerical method, an analytic solution for the temperature in the conductive limit is used; this limit corresponds to $Ma \to 0$ (for which the time scale used in nondimensionalizing is inappropriate). If the unit surface heat input were concentrated at a single point, then the conductive solution would have spherical symmetry:

$$T(r,z) = \frac{1}{2\pi R} \quad , \quad R \equiv \sqrt{r^2 + z^2} \tag{40}$$

For a distributed (axisymmetric) heat source $q(r)$, the point source solution (40) can be used as a Green's function, and the solution found by superposition:

$$T(r,z) = \int_0^\infty \int_0^{2\pi} \frac{q(\rho)\,\rho\,d\theta\,d\rho}{2\pi\sqrt{\rho^2+r^2-2\rho r\cos\theta+z^2}} \tag{41}$$

$$= \int_0^\infty \frac{q(\rho)\,\rho}{\sqrt{(\rho+r)^2+z^2}}\,{}_2F_1(\tfrac{1}{2},\tfrac{1}{2};1;\frac{4\rho r}{(\rho+r)^2+z^2})\,d\rho \tag{42}$$

where ${}_2F_1$ is the generalized hypergeometric function (see [1]). This formula can be used to find the temperature for any input heating distribution q, and the isotherm $T=1$ specifies the interface position.

Using this thermal solution with the interface position fixed, the flow equations (31)–(38) are solved numerically in the viscous limit $Re \to 0$ (again, the time scale used is inappropriate in this limit). This gives the basic state, which has no fine details (except near the concentrated heating, where the flow can be described by an asymptotic solution [3]). This state is used as a starting point for solutions with low Ma and high Pr.

NUMERICAL METHODS

For computational purposes, the idealized problem of an unbounded solid is truncated to a finite domain in cylindrical coordinates, extending in both the radial and vertical directions a distance of four times the diffusion length scale d. The boundary condition on this artificial boundary is that the temperature should decay in the same way as the conduction solution for the point source, that is,

$$\frac{\partial T}{\partial R} = -\frac{T}{R} \tag{43}$$

where $R=\sqrt{r^2+z^2}$ is the spherical coordinate. This asymptotic matching condition is reasonable (for several diffusion lengths away from the pool) and is far less restrictive than imposing the Dirichlet condition ($T=0$) on the outer boundary.

To calculate the steady state for various values of Ma and Pr, the time-dependent equations are stepped in time using the Crank-Nicholson method to obtain the advantages of absolute stability and large time steps. Then at each time step, an elliptic problem must be solved. For this, multilevel methods are used, based on a uniform grid in the (r,z) quarter-plane and the Fast Adaptive Composite (FAC) grid approach to ensure resolution of all small-scale local details. At the solid-liquid interface, each grid has irregular elements to fit the interface. At each time step, the position of the interface is adjusted based on the normal velocity V from (30). (Note that the dimensionless parameter λ in (30) can be adjusted to control how quickly the interface changes.) The difference equations on the grid are developed using the Finite Volume Element (FVE) method. This method combines the exact conservation of mass, momentum, and energy of the finite volume method with the flexibility of the

finite element method in handling complicated boundary conditions, irregular grids, etc. (See [5] for an introduction to FAC and FVE methods.) The resulting system of algebraic equations is solved at each time step. FAC is a method in which the solutions at the various grid levels are used to correct the composite grid solution, and the type of solver used on each grid level is unimportant. In this work both direct methods and iterative solution by line relaxation are used as solvers at each grid level.

FVE STENCILS

To recapitulate, the complete system of dimensionless equations is

$$\text{solid:} \qquad \frac{\partial T}{\partial t} = \frac{1}{Ma}\nabla\cdot\nabla T \tag{44}$$

$$\text{liquid:} \qquad \frac{\partial T}{\partial t} + \nabla\cdot(\mathbf{u}T) = \frac{1}{Ma}\nabla\cdot\nabla T \tag{45}$$

$$\frac{\partial \omega}{\partial t} - \nabla\times(\mathbf{u}\times\omega) = -\frac{1}{Re}\nabla\times\nabla\times\omega \tag{46}$$

$$\omega = \nabla\times\nabla\times\left(\frac{\Psi}{r}\hat{\theta}\right) \tag{47}$$

$$\text{where} \qquad \mathbf{u} = \nabla\times\left(\frac{\Psi}{r}\hat{\theta}\right) = -\frac{1}{r}\frac{\partial\Psi}{\partial z}\hat{\mathbf{r}} + \frac{1}{r}\frac{\partial\Psi}{\partial r}\hat{\mathbf{z}} \tag{48}$$

with the boundary conditions

$$\text{solid surface } z=0 \;:\; \frac{\partial T}{\partial z} = 0 \tag{49}$$

$$\text{liquid surface } z=0 \;:\; \frac{\partial T}{\partial z} = -q(r) \tag{50}$$

$$\Psi = 0 \tag{51}$$

$$\omega = \frac{\partial T}{\partial r}\hat{\theta} \tag{52}$$

$$\text{axis } r=0 \;:\; \frac{\partial T}{\partial r} = 0 \tag{53}$$

$$\Psi = 0 \tag{54}$$

$$\omega = \mathbf{0} \tag{55}$$

$$\text{far away } r,z\to\infty \;:\; \frac{\partial T}{\partial R} \to -\frac{T}{R} \quad \left(\text{where } R\equiv\sqrt{r^2+z^2}\right) \tag{56}$$

$$\text{interface } r=f(z,t) \;:\; T = 1 \tag{57}$$

$$\Psi = \frac{\partial\Psi}{\partial n} = 0 \tag{58}$$

$$V_n = \lambda^{-1}\left[\left(\frac{\partial T}{\partial n}\right)_s - \left(\frac{\partial T}{\partial n}\right)_l\right] \tag{59}$$

where n refers to the direction normal to the interface (outward). (Note: $\int_0^\infty q(r)\,r\,dr = 1$.)

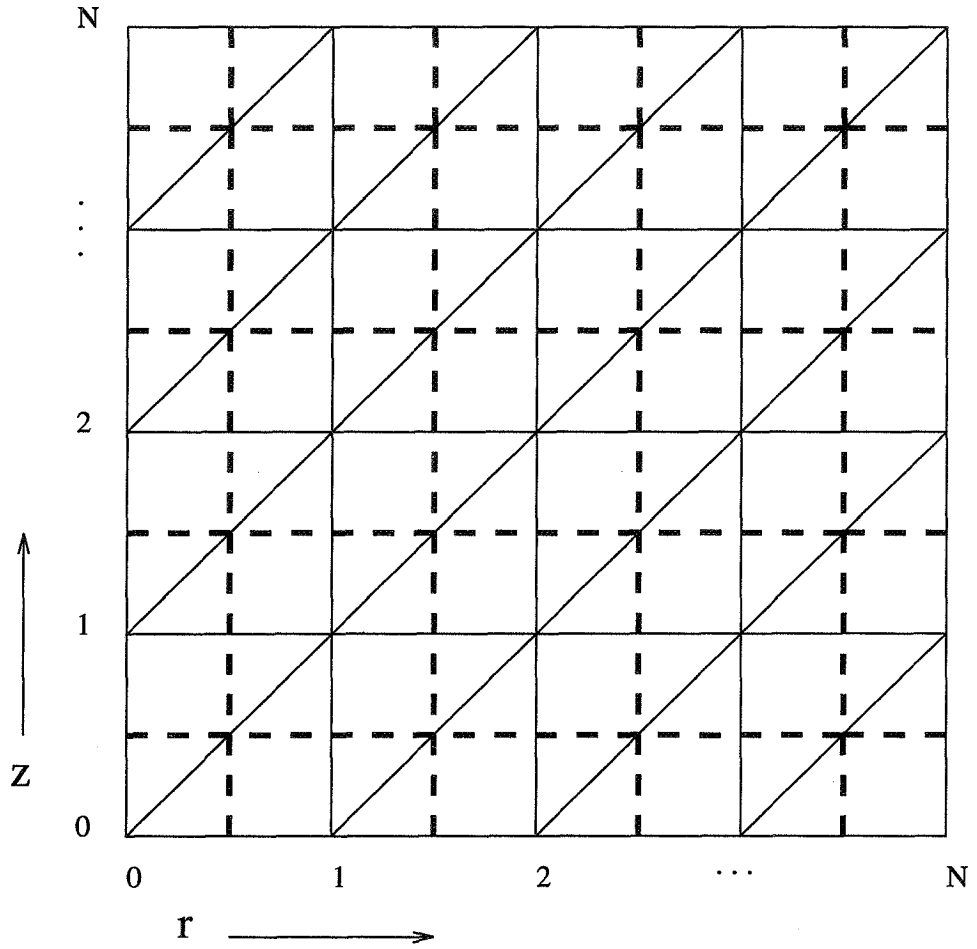


Figure 2: *FVE Grid: the orientation of the triangular finite elements (solid) and the square finite volumes (dashed) are shown. On each triangular element, the variables are assumed linear between the three nodes. This allows consistent calculation of the gradients across the volume boundaries. Note that this is only a cross section in the* (r, z) *plane; the volumes extend in the* θ *direction to form rings.*

The Finite Volume Element (FVE) approach to discretizing the system involves decomposing the domain in two ways: as the union of a set of elements, whose vertices compose the set of grid points on which the unknowns are defined; and as the union of a set of control volumes, one for each grid point (see Figure 2). The unknowns are interpolated over each element, based on the values at the grid points, giving a continuous representation over the whole domain. This representation is used to integrate the conservation equations over each control volume. Hence, each control volume gives three equations involving the three unknowns at the associated grid point, as well as the values at neighboring points. The resulting set of discrete equations for the finite element representation of the solution satisfies the conservation laws exactly over any volume made up of the union of control volumes, including the whole domain. (Actually, the boundary conditions may eliminate some of the control volumes.)

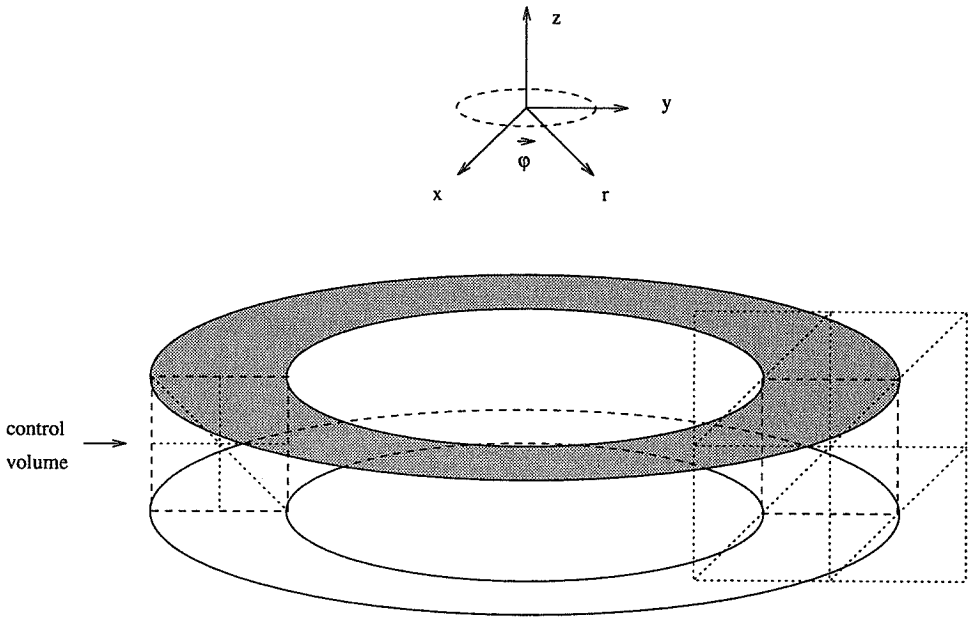


Figure 3: *From axisymmetry, each control volume results from sweeping the square cross-section in the (r, z) plane about the z axis, giving a toroidal prism shape. Hence, the uniform grid gives control volumes that increase with radial position.*

For this axisymmetric problem, each control volume is a toroidal prism, the result of taking a polygonal cross-section in the (r, z) plane and sweeping it all the way around in the θ direction (see Figure 3). Then, integrating the convection-diffusion equation (45) over a control volume, interchanging time derivatives and spatial integrals, and applying the divergence theorem gives

$$\frac{d}{dt} \iint_A T\, r\, dr\, dz + \oint_C \hat{\mathbf{n}} \cdot (\mathbf{u}T)\, r\, dl = \frac{1}{Ma} \oint_C \hat{\mathbf{n}} \cdot \nabla T\, r\, dl \tag{60}$$

where the 2π resulting from integration in θ has been factored out, A refers to the cross-sectional area (polygon) of the volume, C refers to the closed curve bounding that cross-section, and $\hat{\mathbf{n}}$ is the unit vector normal (outward) to C.

For the vorticity (46) and stream function equations (47), the control volume is a vorticity tube, and the appropriate integral is over the cross-sectional area A (with normal vector $\hat{\theta}$). Then, applying Stokes' theorem gives

$$\frac{d}{dt} \iint_A \omega \cdot \hat{\theta}\, dr\, dz - \oint_C \hat{\mathbf{t}} \cdot (\mathbf{u} \times \omega)\, dl = -\frac{1}{Re} \oint_C \hat{\mathbf{t}} \cdot \nabla \times \omega\, dl \tag{61}$$

$$\oint_C \hat{\mathbf{t}} \cdot \mathbf{u}\, dl = \iint_A \omega \cdot \hat{\theta}\, dr\, dz \tag{62}$$

where $\hat{\mathbf{t}}$ is the unit vector tangent to C, in the positive θ sense.

Except near the phase-change interface, a uniform grid is applied with step size h in both the r and z directions (see Figure 2). (Portions of this grid may be subdivided into smaller uniform grids by the FAC method.) Each square of the grid is divided

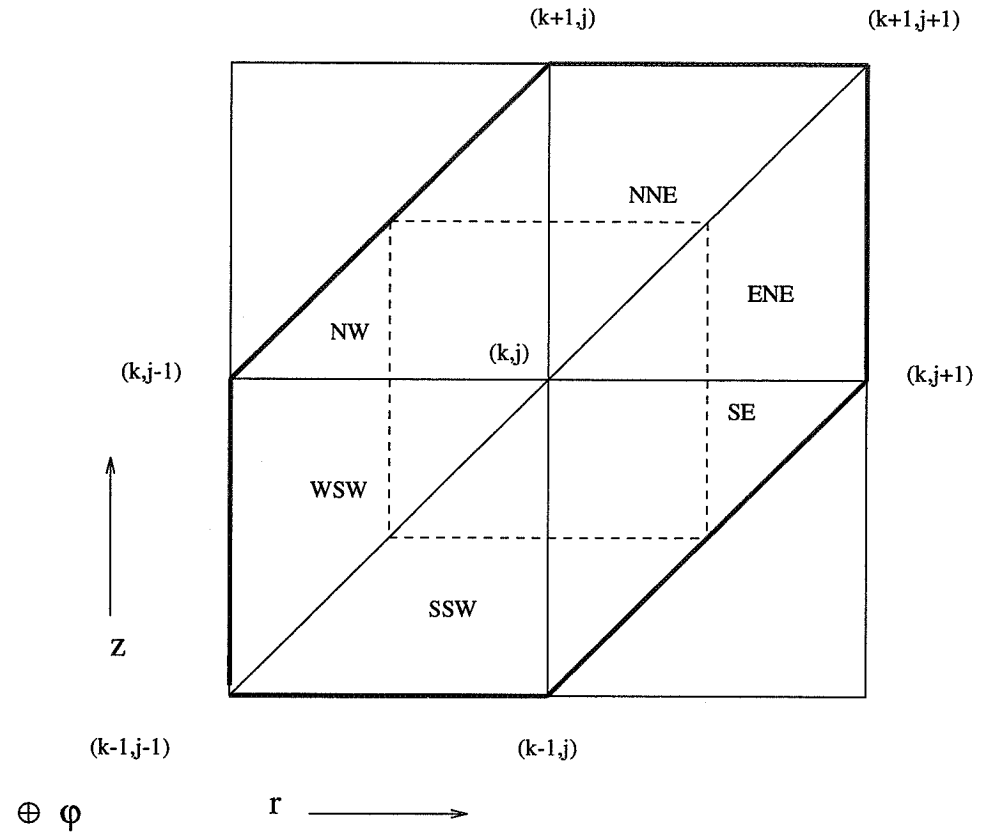


Figure 4: *The conservation integrals for each control volume cross-section involve six separate area integrals over the six triangular elements adjoining the central point; the line integrals involve eight separate parts (the NW and SE elements each contain two segments).*

into two triangular elements by a diagonal (in the direction of increasing $r + z$), and linear interpolation is used over each triangular element. The control volume cross sections are squares of side h, centered on each grid point (except for half-squares at the boundaries and small quarter-squares at the corners).

Then in the integrated conservation equations (60, 61, 62), the area integrals are over six triangular regions (portions of the six elements), and the line integrals are over four line segments, each with halves in two different elements (see Figure 4). In terms of components, the integrated equations are

$$\begin{aligned} \frac{d}{dt}\iint_A T\, r\, dr\, dz \;+\; & \int_N \frac{\partial \Psi}{\partial r} T\, dr - \int_E \frac{\partial \Psi}{\partial z} T\, dz - \int_S \frac{\partial \Psi}{\partial r} T\, dr + \int_W \frac{\partial \Psi}{\partial z} T\, dz \\ = \; & \frac{1}{Ma}\left(\int_N \frac{\partial T}{\partial z} r\, dr + \int_E \frac{\partial T}{\partial r} r\, dz - \int_S \frac{\partial T}{\partial z} r\, dr - \int_W \frac{\partial T}{\partial r} r\, dz \right), \end{aligned} \tag{63}$$

$$\frac{d}{dt}\iint_A \omega\, dr\, dz \;+\; \int_N \frac{\partial \Psi}{\partial r}\frac{\omega}{r} dr - \int_E \frac{\partial \Psi}{\partial z}\frac{\omega}{r} dz - \int_S \frac{\partial \Psi}{\partial r}\frac{\omega}{r} dr + \int_W \frac{\partial \Psi}{\partial z}\frac{\omega}{r} dz$$

$$= \frac{1}{Re}\left(\int_N \frac{\partial \omega}{\partial z}\,dr + \int_E \frac{\partial (r\,\omega)}{\partial r}\frac{1}{r}\,dz - \int_S \frac{\partial \omega}{\partial z}\,dr - \int_W \frac{\partial (r\,\omega)}{\partial r}\frac{1}{r}\,dz\right), \tag{64}$$

$$-\iint_A \omega\,dr\,dz = \int_N \frac{\partial \Psi}{\partial z}\frac{1}{r}\,dr + \int_E \frac{\partial \Psi}{\partial r}\frac{1}{r}\,dz - \int_S \frac{\partial \Psi}{\partial z}\frac{1}{r}\,dr - \int_W \frac{\partial \Psi}{\partial r}\frac{1}{r}\,dz, \tag{65}$$

where from here onward, ω refers to the one nonzero component of vorticity, and the labels N, E, S, and W refer to the four line segments of the line integrals by "compass direction" relative to the central node.

Substituting the piecewise linear element representation of the unknowns into the above integrals gives the discrete (in space) equations. (We use *Maple* to evaluate and sum the integrals for these equations.) The equations are then presented in stencil notation. In stencil notation, for example

$$\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} T,$$

where the center of the matrix e is the coefficient of T at the gridpoint P, the other entries (a, $b\ldots$) in the matrix are the coefficients of the values the unknown (T) at the neighboring gridpoints, and r and z are horizontal and vertical coordinates of P, respectively. Blank entries indicate zero coefficients, and a central Σ indicates the sum of all the other coefficients in the matrix. Note that in the nonlinear convective terms, each of the coefficients of T or ω is itself expressed as a stencil in Ψ (each centered at the same point P); to save space, the Ψ is left out of the vorticity convection stencil.

At a typical grid point, the discretized equations become

$$\frac{d}{dt}\frac{h^2}{24}\begin{pmatrix} & 2-\frac{5}{16}\epsilon & 1+\frac{5}{16}\epsilon \\ 2-\frac{11}{16}\epsilon & 14 & 2+\frac{11}{16}\epsilon \\ 1-\frac{5}{16}\epsilon & 2+\frac{5}{16}\epsilon & \end{pmatrix} T$$

$$+\frac{1}{8r}\begin{pmatrix} & & \begin{pmatrix} & & 1 \\ -2 & 1 & \end{pmatrix}\Psi & \begin{pmatrix} & -1 & \\ & & 1 \end{pmatrix}\Psi \\ \begin{pmatrix} & 2 & \\ & -1 & \\ -1 & & \end{pmatrix}\Psi & \begin{pmatrix} & -1 & \\ 1 & & 1 \\ & -1 & \end{pmatrix}\Psi & \begin{pmatrix} & & -1 \\ & -1 & \\ 2 & & \end{pmatrix}\Psi \\ \begin{pmatrix} 1 & & \\ & -1 & \end{pmatrix}\Psi & \begin{pmatrix} & 1 & -2 \\ 1 & & \end{pmatrix}\Psi & \end{pmatrix} T$$

$$= \frac{1}{Ma}\begin{pmatrix} & 1 & \\ 1-\frac{1}{2}\epsilon & -4 & 1+\frac{1}{2}\epsilon \\ & 1 & \end{pmatrix} T \tag{66}$$

$$\frac{d}{dt}\frac{h^2}{24}\begin{pmatrix} & 2 & 1 \\ 2 & 14 & 2 \\ 1 & 2 & \end{pmatrix}\omega + \frac{1}{8r}\begin{pmatrix} & [A_1]\Psi & [A_2]\Psi \\ [A_3]\Psi & [C_A]\Psi & [A_4]\Psi \\ [A_5]\Psi & [A_6]\Psi & \end{pmatrix}\omega$$
$$= \frac{1}{Re}\begin{pmatrix} & 1-\frac{1}{8}\epsilon^+ & \frac{1}{8}\epsilon^- \\ 1-\frac{3}{8}\epsilon^+ & -\left[4+\frac{3}{8}(\epsilon^+-\epsilon^-)\right] & 1+\frac{3}{8}\epsilon^- \\ -\frac{1}{8}\epsilon^+ & 1+\frac{1}{8}\epsilon^- & \end{pmatrix}\omega \tag{67}$$

$$\begin{pmatrix} & 1 & \\ 1^+ & -[2+1^+ +1^-] & 1^- \\ & 1 & \end{pmatrix}\Psi = -\frac{rh^2}{24}\begin{pmatrix} & 2 & 1 \\ 2 & 14 & 2 \\ 1 & 2 & \end{pmatrix}\omega \tag{68}$$

where the internal stencils

$$[C_A]\Psi \equiv \begin{pmatrix} & -1+\frac{1}{2}\epsilon^+ & \epsilon^- \\ 1+\epsilon^+ & -\epsilon^+ +\epsilon^- & 1-\epsilon^- \\ -\epsilon^+ & -1-\frac{1}{2}\epsilon^- & \end{pmatrix}\Psi,$$

$$[A_1] \equiv \begin{pmatrix} & \frac{1}{2}\epsilon^+ & 1 \\ -2 & 1-\frac{1}{2}\epsilon^+ & \end{pmatrix}, \qquad [A_2] \equiv \begin{pmatrix} & -1 & \frac{1}{2}\epsilon^- \\ & & 1^- \\ & & \end{pmatrix},$$

$$[A_3] \equiv \begin{pmatrix} & 2+1^+ & \\ \frac{1}{2}\epsilon^+ & -1-\epsilon^+ & \\ -1^+ & & \end{pmatrix}, \qquad [A_4] \equiv \begin{pmatrix} & & -1^- \\ & -1+\epsilon^- & -\frac{1}{2}\epsilon^- \\ & 2+1^- & \end{pmatrix},$$

$$[A_5] \equiv \begin{pmatrix} & 1^+ & \\ -\frac{1}{2}\epsilon^+ & -1 & \end{pmatrix}, \qquad [A_6] \equiv \begin{pmatrix} & 1+\frac{1}{2}\epsilon^- & -2 \\ 1 & -\frac{1}{2}\epsilon^- & \end{pmatrix},$$

and the definitions

$$\epsilon \equiv h/r, \quad \epsilon^+ \equiv \frac{\epsilon}{1-\epsilon/2}, \quad \epsilon^- \equiv \frac{\epsilon}{1+\epsilon/2}, \quad 1^+ \equiv \frac{1}{1-\epsilon/2}, \quad 1^- \equiv \frac{1}{1+\epsilon/2},$$

are employed and r is the radial coordinate at the central point P. Note that for those integrals in r with integrands containing $1/r$, that factor was pulled outside the integral to avoid logarithms; the error introduced is of the same order as that due to the piecewise linear representation itself. Also, the heat equation was rescaled by $1/r$, and the stream function equation was rescaled by r.

The radial dependence of the coefficients is a direct result of the axisymmetric geometry. This dependence makes the calculation somewhat more complicated than the corresponding two-dimensional problem. But far from the axis, where $r \gg h$ and hence $\epsilon \ll 1$, the equations approach the corresponding two-dimensional versions, facilitating comparison.

Discretized Boundary Conditions

Along the surface $z = 0$, each of the three boundary conditions for the three unknowns requires different treatment. The temperature at each grid point along the surface is determined by a heat balance over the corresponding control volume, with a half-square cross section $(h \times h/2)$. The contribution of the surface to the convective flux integral is zero, since there is no velocity normal to the surface, and the contribution of the surface to the diffusive flux integral is given by the Neumann type boundary condition $\int q(r) r\, dr$. The resulting discrete equation is

$$\frac{d}{dt}\frac{h^2}{24}\begin{pmatrix} & 2-\frac{5}{16}\epsilon & 1+\frac{5}{16}\epsilon \\ \frac{3}{2}-\frac{1}{2}\epsilon & 7+\frac{5}{16}\epsilon & \frac{1}{2}+\frac{3}{16}\epsilon \end{pmatrix} T$$

$$+\ \frac{1}{8r}\begin{pmatrix} & & \begin{pmatrix} & 1 \\ & \end{pmatrix}\Psi & \begin{pmatrix} & -1 \\ & \end{pmatrix}\Psi \\ \begin{pmatrix} & 2 \\ & \end{pmatrix}\Psi & \begin{pmatrix} & -1 \\ & \end{pmatrix}\Psi & \begin{pmatrix} & -1 \\ & \end{pmatrix}\Psi \end{pmatrix} T$$

$$=\ \frac{1}{Ma}\begin{pmatrix} & 1 & \\ \frac{1}{2}-\frac{1}{4}\epsilon & -2 & \frac{1}{2}+\frac{1}{4}\epsilon \end{pmatrix} T + \frac{1}{r\,Ma}\int_{r-h/2}^{r+h/2} q(r) r\, dr \qquad (69)$$

Here we specify the heat flux as a symmetric function of r that decays smoothly to zero at some finite radius ρ_{max}, while satisfying $\int_0^\infty q(r) r\, dr = 1$:

$$q(r) \equiv \begin{cases} \frac{6}{\rho_{max}^2}\left[1-\left(\frac{r}{\rho_{max}}\right)^2\right]^2, & r \le \rho_{max} \\ 0, & r > \rho_{max} \end{cases} \qquad (70)$$

For the calculations, we use $\rho_{max} = \frac{1}{4}$.

The thermocapillary stress condition at the surface specifies the vorticity: $\omega = \frac{\partial T}{\partial r}$. However, because of the linear interpolation between grid points, $\frac{\partial T}{\partial r}$ is not well defined at grid points on the surface. Hence, for the surface only, the vorticity is specified at half-grid points (i.e., $r = (i + \frac{1}{2})h$), and triangular finite elements are formed with neighboring points. This keeps the discretization of this important condition at the same order of accuracy as the other equations, but entails special treatment of the grid points next to the surface. The surface is also a streamline, where $\Psi = 0$ (Dirichlet condition). Using that fact and these special surface vorticity elements gives the following flow equations for points by the surface (a distance h from the surface):

$$\frac{d}{dt}\frac{h^2}{24}\begin{pmatrix} \frac{19}{8} & 2 & 1 \\ & 14\frac{1}{4} & \frac{11}{8} \end{pmatrix}\omega + \frac{d}{dt}\frac{h}{16}\begin{pmatrix} & \\ -1 & 1 \end{pmatrix} T + \frac{1}{8r}\begin{pmatrix} & [B_1]\Psi & [B_2]\Psi \\ [B_3]\Psi & [C_B]\Psi & [B_4]\Psi \end{pmatrix}$$

$$+\frac{1}{8rh}\begin{pmatrix} & & \\ [D_1] & [D_2] & [D_3] \end{pmatrix} T = \frac{1}{Re}\begin{pmatrix} & 1-\frac{1}{8}\epsilon^+ & \frac{1}{8}\epsilon^- \\ \frac{7}{8}-\frac{7}{16}\epsilon^+ & -\left[\frac{15}{4}+\frac{5}{16}\epsilon^+-\frac{7}{16}\epsilon^-\right] & \frac{7}{8}+\frac{5}{16}\epsilon^- \end{pmatrix}\omega$$

$$+\frac{1}{Re}\frac{1}{h}\begin{pmatrix} & & \\ -\frac{1}{2}+\frac{1}{8}\epsilon^+ & -\frac{1}{8}\epsilon^+-\frac{1}{8}\epsilon^- & \frac{1}{2}+\frac{1}{8}\epsilon^- \end{pmatrix} T \quad (71)$$

$$\begin{pmatrix} & 1 & \\ 1^+ & -[2+1^+ +1^-] & 1^- \end{pmatrix}\Psi = \frac{rh^2}{24}\begin{pmatrix} & 2 & 1 \\ \frac{19}{8} & 14\frac{1}{4} & \frac{11}{8} \end{pmatrix}\omega + \frac{rh}{16}\begin{pmatrix} & \\ -1 & 1 \end{pmatrix} T \quad (72)$$

where

$$[C_B] \equiv \begin{pmatrix} & -1+\frac{1}{2}\epsilon^+ & \epsilon^- \\ \frac{1}{2}+\frac{3}{4}\epsilon^+ & 1-\frac{1}{2}\epsilon^+ +\frac{3}{4}\epsilon^- & \frac{1}{4}-\epsilon^- \end{pmatrix}$$

$$[B_1] \equiv \begin{pmatrix} & \frac{1}{2}+\frac{3}{4}\epsilon^+\epsilon^+ & 1 \\ -2 & 1-\frac{1}{2}\epsilon^+ & \end{pmatrix}, \quad [B_2] \equiv \begin{pmatrix} -1 & \frac{1}{2}\epsilon^- \\ & 1^- \end{pmatrix},$$

$$[B_3] \equiv \begin{pmatrix} & 2+1^+ \\ \frac{1}{2}+\frac{3}{4}\epsilon^+ & -1-\epsilon^+ \end{pmatrix}, \quad [B_4]\begin{pmatrix} & -1^- \\ -\frac{5}{4}+\frac{3}{4}\epsilon^- & \frac{3}{4}-\frac{1}{2}\epsilon^- \end{pmatrix},$$

$$[D_1] \equiv \begin{pmatrix} -1^+ & -\frac{1}{4} & \frac{1}{4} \end{pmatrix}, \quad [D_2] \equiv \begin{pmatrix} 1^+ & -\frac{1}{2}-\frac{1}{2}\epsilon^+ & \frac{3}{2} \end{pmatrix},$$

$$\text{and} \quad [D_3] \equiv \begin{pmatrix} \frac{3}{4}+\frac{1}{2}\epsilon^+ & -\frac{7}{4} \end{pmatrix}$$

Along the z axis, symmetry requires that there is no heat flux across the axis, nor flow, nor shear stress, so both Ψ and ω are zero there. Then for points on the axis, the discrete heat balance over the cylindrical control volumes (half-square cross section $h/2 \times h$) gives:

$$\frac{d}{dt}\frac{h^2}{24}\begin{pmatrix} \frac{1}{4} & \frac{5}{4} \\ \frac{25}{4} & \frac{11}{4} \\ \frac{6}{4} & \end{pmatrix} T + \frac{1}{2h}\begin{pmatrix} \begin{pmatrix} & 1 \end{pmatrix}\Psi & \begin{pmatrix} 1 & \end{pmatrix}\Psi \\ \begin{pmatrix} & 1 \end{pmatrix}\Psi & \begin{pmatrix} & -1 \\ & \end{pmatrix}\Psi \\ \begin{pmatrix} & -2 \end{pmatrix}\Psi & \end{pmatrix} T$$

$$-\frac{1}{Ma}\begin{pmatrix} & \frac{1}{2} & \\ & -3 & 2 \\ & \frac{1}{2} & \end{pmatrix} T \;=\; 0 \quad (73)$$

where the equation was scaled using the average $\bar{r} = h/4$. The homogeneous Dirichlet conditions on Ψ and ω apply to points on the axis, and for grid points neighboring the axis, the usual stencils apply; no special treatment is necessary.

The temperature at the grid point at the origin is determined by a small control volume (quarter-square cross section $h/2 \times h/2$) with specified surface heat flux and no flux (nor convection) through the axis:

$$\frac{d}{dt}\frac{h^2}{24}\begin{pmatrix} \frac{1}{4} & \frac{5}{4} \\ \frac{15}{4} & \frac{3}{4} \end{pmatrix} T \;+\; \frac{1}{2h}\begin{pmatrix} \begin{pmatrix} & 1 \\ & \end{pmatrix}\Psi & \\ & \begin{pmatrix} & -1 \\ & \end{pmatrix}\Psi \end{pmatrix} T$$

$$-\frac{1}{Ma}\begin{pmatrix} & \frac{1}{2} & \\ & -\frac{3}{2} & 1 \end{pmatrix} T = \frac{1}{Ma}\frac{4}{h}\int_0^{h/2} q(r) r\, dr \quad (74)$$

Again, at the origin, both Ψ and ω are zero (note the two boundary conditions on vorticity are consistent at this point, due to the symmetry). Hence, the usual surface flow equations apply to the grid point next to the origin.

At the far boundaries of the computational domain, the boundary condition on the heat diffusion equation in the solid is that it decays in the same way as the spherically symmetric solution for a point source:

$$\nabla T = \frac{\partial T}{\partial R}\hat{\mathbf{R}} = -\frac{T}{R}\hat{\mathbf{R}} = -T\frac{r}{R^2}\hat{\mathbf{r}} - T\frac{z}{R^2}\hat{\mathbf{z}} \quad (75)$$

where $R \equiv \sqrt{r^2 + z^2}$. This allow the heat flux across the artificial boundary to be computed in terms of the temperature there, a Robin type boundary condition. Below we give the discrete equations for the two edges (half-square volumes) and three corners (quarter-square volumes) where this boundary condition is applied.

At the edge where r is at its maximum the stencil is given by

$$\frac{d}{dt}\frac{h^2}{24}\begin{pmatrix} & \frac{3}{2}-\frac{3}{8}\epsilon & \\ 2-\frac{11}{16}\epsilon & 7-\frac{25}{16}\epsilon & \\ 1-\frac{5}{16}\epsilon & \frac{1}{2}-\frac{1}{16}\epsilon & \end{pmatrix} T - \frac{1}{Ma}\begin{pmatrix} & \frac{1}{2}-\frac{1}{8}\epsilon-\frac{1}{8}\rho & \\ 1-\frac{1}{2}\epsilon & -2+\frac{3}{4}\epsilon-\frac{3}{4}\rho & \\ & \frac{1}{2}-\frac{1}{8}\epsilon-\frac{1}{8}\rho & \end{pmatrix} T = 0$$

where $\rho \equiv hr/(r^2+z^2)$.

At the edge where z is at its maximum the stencil is

$$\frac{d}{dt}\frac{h^2}{24}\begin{pmatrix} \frac{1}{2}-\frac{3}{16}\epsilon & 7-\frac{5}{16}\epsilon & \frac{3}{2}+\frac{1}{2}\epsilon) \\ 1-\frac{5}{16}\epsilon & 2+\frac{5}{16}\epsilon & \end{pmatrix} T - \frac{1}{Ma}\begin{pmatrix} \alpha & \beta & \gamma \\ & 1 & \end{pmatrix} T = 0$$

where

$$\alpha \equiv \frac{1}{2} - \frac{1}{4}\epsilon - \frac{1}{8}(1 - \frac{1}{3}\epsilon)\zeta, \quad \beta \equiv -2 - \frac{3}{4}\zeta, \quad \gamma \equiv \frac{1}{2} + \frac{1}{4}\epsilon - \frac{1}{8}(1 + \frac{1}{3}\epsilon)\zeta,$$

and $\zeta \equiv hz/(r^2 + z^2)$.

At the corner where both r and z are at a maximum the stencil is

$$\frac{d}{dt}\frac{h^2}{24}\begin{pmatrix} \frac{1}{2} - \frac{3}{16}\epsilon & 4 - \frac{15}{16}\epsilon \\ 1 - \frac{5}{16}\epsilon & \frac{1}{2} - \frac{1}{16}\epsilon \end{pmatrix} T - \frac{1}{Ma}\begin{pmatrix} \mu & \nu \\ & \xi \end{pmatrix} T = 0$$

where

$$\mu \equiv \frac{1}{2} - \frac{1}{4}\epsilon - \frac{1}{8}(1 - \frac{1}{3}\epsilon), \quad \nu \equiv -1 + \frac{3}{8}\epsilon - \frac{3}{8}[\rho + (1 - \frac{2}{9}\epsilon)\zeta],$$

and

$$\xi \equiv \frac{1}{2} - \frac{1}{8}\epsilon - \frac{1}{8}\rho.$$

At the corner where $r = 0$ and z is at a maximum we have the stencil

$$\frac{d}{dt}\frac{h^2}{24}\begin{pmatrix} \frac{5}{2} & 2 \\ \frac{3}{2} & \end{pmatrix} T - \frac{1}{Ma}\begin{pmatrix} -\frac{3}{2} - \frac{1}{3}\zeta & 1 - \frac{1}{6}\zeta \\ \frac{1}{2} & \end{pmatrix} T = 0.$$

Finally, at the corner where r is maximum and $z = 0$ the stencil is

$$\frac{d}{dt}\frac{h^2}{24}\begin{pmatrix} & \frac{3}{2} - \frac{3}{8}\epsilon \\ \frac{3}{2} - \frac{1}{2}\epsilon & 3 - \frac{5}{8}\epsilon \end{pmatrix} T - \frac{1}{Ma}\begin{pmatrix} & \frac{1}{2} - \frac{1}{8}\epsilon - \frac{1}{8}\rho \\ \frac{1}{2} - \frac{1}{4}\epsilon & -1 + \frac{3}{8}\epsilon - \frac{3}{8}\rho \end{pmatrix} T = 0.$$

Tracking the Phase-Change Interface

One of the biggest challenges in models of phase change is the tracking over time of the position of the two-phase interface. As one of the main goals of the current research is the examination of the effects of thermocapillary convection on the interface shape, great care is necessary in accurately modeling the geometry and dynamics of the phase change process.

The grid structure must be modified near the interface. (While it would be possible to quantize the interface position to lie on grid points, that would make moving the interface difficult and would introduce errors that would be magnified in the multilevel representation.) We represent the interface as piecewise linear between the points at which it crosses the diagonals of the main grid, which have slopes equal to 1. This representation assumes that the interface orientation never reaches an angle of 135° (or −45°) relative to the surface (i.e., parallel to the main diagonals); this seems reasonable, considering the interface is an isotherm that meets the surface at 90° and

must end at 0° at the axis. (A more general approach would include representations for several different local grid orientations.)

The movement of the interface through melting or solidification is governed by the local heat balance near the interface. Hence the main requirement for the control volume around each interface point (along the diagonals) is that the volume contain the interface both at the current time and at the next time step, that is, the control volumes must allow room for movement. (Then for the next time step, new control volumes may be used.) Hence, not only the current interface position, but also an estimate of the future position, is required to construct the current local grid. An alternate approach is to adjust the solidification timescale parameter λ at each time step to constrain the maximum motion of the interface to remain within the interface control volumes; physically this would correspond to time-dependent latent heat L.

To keep the geometry as simple as possible while allowing the interface points to move along the main diagonals, we construct the control volumes on a diagonal grid. (Here we refer to the control volumes by their cross sections in the (r, z) plane.) The main diagonals are spaced a distance $h/\sqrt{2}$ apart, and control volume boundaries in that direction lie midway between them. Control volume boundaries in the perpendicular direction are spaced the same, unless such boundary would cross the current or predicted interface, in which case that segment is removed, giving a double-wide volume $(\sqrt{2}h \times h/\sqrt{2})$. [Note: it is conceivable that, if the interface orientation exceeds 90°, triple-wide control volumes may be necessary.] Then any grid points within the interface control volumes are removed. If space remains between the interface control volume and the remaining regular grid, an auxiliary grid point is inserted on the diagonal a distance $h/\sqrt{2}$ from the regular grid point, with its diagonal square control volume $(h/\sqrt{2} \times h/\sqrt{2})$. [Note: to simplify the programming, the auxiliary points could be omitted; then the control volumes for the interface points will be either single width (no grid point removed) or triple width (one grid point removed).] Then the control volumes for the regular grid points adjoining this diagonal grid are pentagons in one of three configurations: at an "inside" corner with one diagonal side, two regular sides, and two regular half-sides; at a straight edge, either horizontal or vertical, with one regular side, two regular half-sides, and two diagonal sides; or, at an "outside" corner with three diagonal sides and two regular half-sides.

The auxiliary grid points form triangular elements with neighboring regular grid points and/or neighboring auxiliary grid points. This leaves trapezoidal elements adjoining the interface. Note that triangulating these trapezoids could result in very complicated relations between elements and volumes. Therefore we use a "warped" bilinear interpolation on these trapezoidal elements.

Where the interface intersects the surface or the axis, the grid must be further modified to track these important points. This involves computing the heat balance on a diagonal surface (or axis) control volume and tracking the position of the interface along the diagonal. Depending on the proximity of the interface point on the diagonal to the surface (or axis), then either the interface is extrapolated from the point inside the surface perpendicularly to the surface or the "interface point" on the diagonal

outside the surface is used to linearly interpolate the interface to the surface.

The interface is defined as the isotherm where $T = 1$, and on the interface the fluid velocity is zero (no slip), and so $\Psi = 0$. The unknown vorticity at interface points is determined by the stream function equation integrated over the liquid portion of the control volume; here the circulation can be calculated (with no contribution along the interface, due to no slip) to find the unknown strength of the vorticity tube:

$$\iint_{A_l} \omega \, dr \, dz = \oint_{C_l} \hat{\mathbf{t}} \cdot \mathbf{u} \, dl \tag{76}$$

where A_l is the liquid area, with bounding curve C_l. (Note that this equation contains no time derivative.)

The only remaining unknown is the future position (along the diagonal) of the interface point. This is governed by the heat balance over the liquid and solid portions of the control volume:

$$\begin{aligned} \lambda \, Ma^{-1} \frac{d}{dt} \iint_{A_l} r \, dr \, dz \;\; = \;\; & \\ & -\frac{d}{dt} \iint_{A_l + A_s} T \, r \, dr \, dz \\ & - \oint_{C_l} \hat{\mathbf{n}} \cdot (\mathbf{u} T) \; r \, dl \\ & + Ma^{-1} \oint_{C_l + C_s} \hat{\mathbf{n}} \cdot \nabla T \, r \, dl \end{aligned} \tag{77}$$

where $A_l + A_s$ indicates the entire control volume, with bounding curve $C_l + C_s$. (Note that A_l and C_l vary over the time step, while the control volume as a whole does not.) The interpretation is that the heat coming in by convection and diffusion goes to raise the temperature inside (though the interface temperature is fixed) and to melt some solid, increasing the liquid portion of the volume (the first term).

The discrete equations are very complicated for regular grid points bordering the diagonal interface grid and for interface points, and so are not reproduced here. To guard against typographical errors, the stencils were derived using the symbolic mathematics capabilities of the *Maple* software ([6]). *Maple* converted the resulting expressions into C language code, which were cut and pasted directly into our simulation code.

The diagonal grid around the interface requires local diagonal coordinates. We call these (x, y), where

$$\begin{array}{ll} x = z + r & \\ y = z - r & \text{so that} \end{array} \qquad \begin{array}{l} r = (x - y)/2 \\ z = (x + y)/2 \end{array} \tag{78}$$

Then the velocity becomes

$$\mathbf{u} = -\frac{\sqrt{2}}{r} \frac{\partial \Psi}{\partial y} \hat{\mathbf{x}} + \frac{\sqrt{2}}{r} \frac{\partial \Psi}{\partial x} \hat{\mathbf{y}} \tag{79}$$

Note that the (x, y) coordinates are scaled down in length by a factor of $\sqrt{2}$ relative to the (r, z) coordinates, and on the diagonal grid the values of the (x, y) coordinates are integer multiples of h (rescaled). In the area integrals, the Jacobian gives $dr\,dz = dx\,dy/2$, but in each of the line integrals, the scaling of the differential is exactly compensated by the scaling of the derivative (with respect to x or y). The slight complication of rescaling is more than offset by the simplification of the algebra; otherwise factors of $\sqrt{2}$ would abound. The bilinear interpolation for the trapezoidal elements is also in terms of the (x, y) coordinates, both to simplify integration with respect to the diagonal coordinates, and to avoid the singular case where the trapezoid is a diagonal perfect square, which cannot be interpolated with a bilinear form in (r, z).

Conclusion

In this preliminary work we have developed a finite volume element method for determining the shape of the weldpool. The governing equations and boundary conditions have been discretized in space, and a time-stepping method can be applied to solve the equations. An FAC method has been devised for resolving the fine details near the moving interface and is being implemented as part of the continuing research.

The basic numerical methods discussed have been implemented in code and tested. A future report will describe the details of the time-stepping, the FAC resolution near the interface, and the numerical results on the total problem.

Acknowledgements

This work was supported by the Office of Naval Research, Materials Division (contract N0001492WR24009).

REFERENCES

[1] Milton Abramowitz and Irene A. Stegun. *Handbook of Mathematical Functions.* Dover Publications, Inc., New York, 1965.

[2] D. Canright. Thermocapillary flow near a cold wall. *Phys. Fluids*, 6:1415–1424, 1994.

[3] C. L. Chan, M. M. Chen, and J. Mazumder. Asymptotic solution for thermocapillary flow at high and low Prandtl numbers due to concentrated surface heating. *J. Heat Transfer*, 110:140–146, 1988.

[4] M. M. Chen. Thermocapillary convection in materials processing. In S. K. Samanta, R. Komandiri, R. McMeeking, M. M. Chen, and A. Tseng, editors, *Interdisciplinary Issues in Materials Processing and Manufacturing*, pages 541–557, ASME, 1987.

[5] Stephen F. McCormick. *Multilevel Adaptive Methods for Partial Differential Equations.* Society for Industrial and Applied Mathematics, Philadelphia, 1989.

[6] Waterloo Maple Software. *Maple V, Release 3.* Waterloo, Ontario, Canada, 1994. (X-windows version for Unix workstations).

QUASI-OPTIMAL SCHWARZ METHODS FOR THE CONFORMING SPECTRAL ELEMENT DISCRETIZATION*

Mario Casarin
Courant Institute of Mathematical Sciences, NYU
New York, NY

SUMMARY

Fast methods are proposed for solving the system $K_N x = b$ resulting from the discretization of self-adjoint elliptic equations in three dimensional domains by the spectral element method. The domain is decomposed into hexahedral elements, and in each of these elements the discretization space is formed by polynomials of degree N in each variable. Gauss-Lobatto-Legendre (GLL) quadrature rules replace the integrals in the Galerkin formulation. This system is solved by the preconditioned conjugate gradients method. The conforming finite element space on the GLL mesh consisting of piecewise Q_1 elements produces a stiffness matrix K_h that is spectrally equivalent to the spectral element stiffness matrix K_N. The action of the inverse of K_h is expensive for large problems, and is therefore replaced by a Schwarz preconditioner B_h of this finite element stiffness matrix. The preconditioned operator then becomes $B_h^{-1} K_N$.

The technical difficulties stem from the nonregularity of the mesh. Tools to estimate the convergence of a large class of new iterative substructuring and overlapping Schwarz preconditioners are developed. This technique also provides a new analysis for an iterative substructuring method proposed by Pavarino and Widlund for the spectral element discretization.

INTRODUCTION

In the past decade, many preconditioners have been developed for the large systems of linear equations arising from the finite element discretization of elliptic self-adjoint partial differential equations; see e.g. [5], [10], [21]. An especially challenging problem is the design of preconditioners for three dimensional problems. More recently, spectral element discretizations of such equations have been proposed, and their efficiency has been demonstrated; see [11], [12], and references therein. In large scale problems,

*This work has been supported in part by a Brazilian graduate student fellowship from CNPq, and in part by the U. S. Department of Energy under contract DE-FG02-92ER25127.

long range interactions of the basis elements produce quite dense and expensive factorizations of the stiffness matrix, and the use of direct methods is not economical due to the large memory requirements [9].

Early work on preconditioners for these equations was done by Pavarino [15],[16], [17]. Some of his algorithms are numerically scalable (i.e., the number of iterations is independent of the number of substructures) and optimal (the number of iterations does not grow or grows slowly with the degree of the polynomials). However, each application of the preconditioner can be very expensive. The bounds for the condition number of the preconditioned operator grow only slowly with the polynomial degrees, and are independent of the number of substructures.

Following Pahl [13], who based his work on the work of Deville and Mund [6] and of Canuto [4], the above constructions give rise to different, spectrally equivalent preconditioners using block partitioning of the *finite element* matrix generated by Q_1 elements on the hexahedrals of the Gauss-Lobatto-Legendre (GLL) mesh. This observation and experiments for a model problem in two dimensions were made by Pahl [13], who demonstrated experimentally that this preconditioner is very efficient. Thus, high order accuracy can be combined with efficient and inexpensive low-order preconditioning. We remark that similar ideas also appear in [20] and references therein, and that the spectral equivalence results of Canuto [4] and generalizations for other boundary conditions were also obtained independently by Parter and Rothman [14].

The previous analysis of Schwarz preconditioners for piecewise linear finite elements for the h-method has relied upon the shape regularity of the mesh [8], [7], [3], which clearly does not hold for the GLL meshes. We extend the analysis to such meshes, deriving estimates for these finite element preconditioners of spectral element methods.

We give polylogarithmic bounds on the condition number of the preconditioned operators for iterative substructuring methods, as well as a new proof of one of the estimates in [18]. We remark that the tools developed here can be used to analyze overlapping Schwarz methods defined on the GLL mesh.

DIFFERENTIAL AND DISCRETE MODEL PROBLEMS

Let Ω be a bounded polyhedral region in $\mathbb{R}^3$ with diameter of order 1. We consider the following elliptic self-adjoint problem:

$$a(u,v) = f(v) \quad \forall\, v \in H_0^1(\Omega), \tag{1}$$

where

$$a(u,v) = \int_\Omega k(x)\, \nabla u \cdot \nabla v \, dx \quad \text{and} \quad f(v) = \int_\Omega f v \, dx \quad \text{for} \quad f \in L^2(\Omega).$$

This problem is discretized by the spectral element method (SEM); see [12]. Namely, we triangulate Ω into nonoverlapping substructures $\{\Omega_i\}_{i=1}^M$ of diameter on the order of H. Each Ω_i is the image of the reference cube $\hat{\Omega} = [-1,+1]^3$ under a mapping $F_i = D_i \circ G_i$ where D_i is an isotropic dilation and G_i is a C^∞ mapping such that its derivative and the inverse of its derivative are uniformly bounded by a constant close to one. Moreover, we suppose that the intersection between the closure of two substructures is either empty, a vertex, a whole edge or a whole face. Each substructure Ω_i is a distorted cube. We notice that some additional properties of the mappings F_i are required to guarantee an optimal convergence rate. We refer to [2], problem 2 and the references therein for further details on this issue, but remark that affine mappings are covered by the available convergence theory for these methods. We assume for simplicity that $k(x)$ has the constant value $k_i > 0$ in the substructure Ω_i, with possibly large jumps occurring only across substructure boundaries. Our estimates for iterative substructuring algorithms are independent of these jumps.

We define the space $P^N(\hat{\Omega})$ as the space of Q_N functions, i.e. polynomials of degree at most N in each of the variables separately. The space $P^N(\Omega_i)$ is the space of functions v_N such that $v_N \circ F_i$ belongs to $P^N(\hat{\Omega})$. The conforming space $P_0^N(\Omega) \subset H_0^1(\Omega)$ is the space of continuous functions the restrictions of which to Ω_i belong to $P^N(\Omega_i)$ for $i = 1, ..., M$.

The discrete L^2 inner product is defined by

$$(u,v)_N = \sum_{i=1}^{K} \sum_{j,k,l=1}^{N} k \cdot (u \circ F_i) \cdot (v \circ F_i) \cdot |J_i|(\xi_j, \xi_k, \xi_l) \cdot \rho_j \rho_k \rho_l, \tag{2}$$

where ξ_j and ρ_j are, respectively, the Gauss-Lobatto-Legendre (GLL) quadrature points and weights in the interval $[-1,+1]$; see [2].

The discrete problem is: find $u_N \in P_0^N(\Omega)$, such that

$$a_Q(u_N, v_N) = (\nabla u_N, \nabla v_N)_N = f(v_N) \quad \forall\, v_N \in P_0^N(\Omega). \tag{3}$$

We choose as basis functions the functions ϕ_j^N of $P_0^N(\Omega)$ that are one at the GLL node j and zero at the other nodes, which gives rise in the standard way to the linear system $K_N x = b$. Note that the mass matrix of this nodal basis generated by the discrete L^2 inner product (2) is diagonal. The analysis of the SEM method just described and experimental evidence show that it achieves very good accuracy for reasonably small N for a wide range of problems; see [2], [12], and references therein. The practical application of this approach for large scale problems, however, depends on fast and reliable solution methods for the system $K_N x = b$. The condition number of K_N is very large even for moderate values of N; see [2]. Our approach is to solve this system by a preconditioned conjugate gradient algorithm. The following low-order discretization is used to define several preconditioners in the next sections.

The GLL points define a triangulation $\mathcal{T}^{\hat{h}}$ of $\hat{\Omega}$ into parallelepipeds, and on this triangulation we define the space $P^{\hat{h}}(\hat{\Omega})$ of continuous piecewise trilinear (Q_1) functions. The spaces $P^h(\Omega_i)$ and $P_0^h(\Omega)$ are defined analogously to $P^N(\Omega_i)$ and $P_0^N(\Omega)$. The finite element discrete problem associated with (1) is: find $u_h \in P_0^h(\Omega)$ such that

$$a(u_h, v_h) = f(v_h) \quad \forall\, v_h \in P_0^h(\Omega). \tag{4}$$

The standard nodal basis $\{\hat{\phi}_j^{\hat{h}}\}$ in $P^{\hat{h}}(\hat{\Omega})$ is mapped by the F_i, $1 \le i \le M$, into a basis for $P_0^h(\Omega)$. This basis also gives rise to a system $K_h x = b$ in the standard way.

We use the following notations: $x \preceq y$, $z \succeq u$, and $v \asymp w$ to express that there are positive constants C and c such that

$$x \le C\,y, \quad z \ge c\,u \quad \text{and} \quad c\,w \le v \le C\,w, \quad \text{respectively.}$$

Here and elsewhere c and C are moderate constants independent of H and N.

Let $\hat{h}$ be the distance between the first two GLL points in the interval $[-1,+1]$; $\hat{h}$ is proportional to $1/N^2$ [2], and the sides $h_i, i = 1,2,3$ of an element K belonging to $\mathcal{T}^{\hat{h}}$ satisfy

$$1/N^2 \preceq h_i \preceq 1/N,$$

depending on the location of K inside $\hat{\Omega}$. The triangulation is therefore not shape regular.

GENERAL SETUP AND SIMPLIFICATIONS

Let $\hat{u}_N$ be a function belonging to $P^N(\hat{\Omega})$, and let $\hat{u}_h = \hat{I}_N^h \hat{u}_N$ be the function of $P^{\hat{h}}(\hat{\Omega})$ for which

$$\hat{u}_h(x_G) = \hat{u}_N(x_G),$$

for every GLL point x_G in $\hat{\Omega}$. Then

$$|\hat{u}_h|^2_{H^1(\hat{\Omega})} \asymp |\hat{u}_N|^2_{H^1(\hat{\Omega})} \asymp a_{\hat{Q}}(\hat{u}_N, \hat{u}_N), \tag{5}$$

and

$$||\hat{u}_h||^2_{L^2(\hat{\Omega})} \asymp ||\hat{u}_N||^2_{L^2(\hat{\Omega})} \asymp (\hat{u}_N, \hat{u}_N)_N, \tag{6}$$

where $a_{\hat{Q}}$ is given by (2) and (3) with $J_i \equiv 1$; see [4] and [2]. We remark that these results and generalizations for other boundary conditions were obtained independently by Parter and Rothman [14]. The basis of these results is the H^1 stability of the interpolation operator at the GLL nodes for functions of $H^1([-1,+1])$, proved by Bernardi and Maday [1], [2].

Consider now a function v defined in a substructure Ω_i with diameter of order H. Changing variables to the reference substructure by $\hat{v}(\hat{x}) = v(F_i(\hat{x}))$ and using simple estimates on the Jacobian of F_i, we obtain

$$\|u\|^2_{L^2(\Omega_i)} \asymp H^d \|\hat{u}\|^2_{L^2(\hat{\Omega})}, \tag{7}$$

and

$$|u|^2_{H^1(\Omega_i)} \asymp H^{d-2} |\hat{u}|^2_{H^1(\hat{\Omega})}, \tag{8}$$

where the dimension d is equal to 1, 2, or 3.

These estimates can be interpreted as spectral equivalence of the stiffness and mass matrices generated by the norms and basis of the discrete spaces introduced above. Indeed, the nodal basis $\{\hat{\phi}^{\hat{h}}_j\}$ is mapped by interpolation at the GLL nodes to a nodal basis of $P^N(\hat{\Omega})$. Then, (5) can be written as

$$\underline{\hat{u}}^T \hat{K}_h \underline{\hat{u}} \asymp \underline{\hat{u}}^T \hat{K}_N \underline{\hat{u}}, \tag{9}$$

where $\underline{\hat{u}}$ is the vector of nodal values of both $\hat{u}_N$ or $\hat{u}_h$, and $\hat{K}_h$ and $\hat{K}_N$ are the stiffness matrices corresponding to $|\cdot|^2_{H^1(\hat{\Omega})}$ and $a_{\hat{Q}}(.,.)$.

Therefore, if $K_h^{(i)}$ and $K_N^{(i)}$ are the stiffness matrices generated by the basis $\{\phi^h_j\}$ and $\{\phi^N_j\}$, respectively, for all nodes j in the closure of Ω_i and by $|\cdot|^2_{H^1(\Omega_i)}$ and $a_{Q,\Omega_i}(\cdot,\cdot)$, then

$$\underline{u}^T K_h^{(i)} \underline{u} \asymp \underline{u}^T K_N^{(i)} \underline{u},$$

where $\underline{u}$ is the vector of nodal values, by (9), (8), and (5). The stiffness matrices K_N and K_h are formed by subassembly [7],

$$\underline{u}^T K_h \underline{u} = \Sigma_i \underline{u}^{(i)T} K_h^{(i)} \underline{u}^{(i)}, \tag{10}$$

for any nodal vector $\underline{u}$, where the $\underline{u}^{(i)}$ are the subvectors of nodal values in $\overline{\Omega_i}$; an analogous expression holds for K_N. These last two relations imply that

$$\underline{u}^T K_h \underline{u} \asymp \underline{u}^T K_N \underline{u}, \tag{11}$$

for any vector $\underline{u}$. All these matrix equivalences and their analogues in terms of norms are hereafter called the FEM-SEM equivalence.

We next show that the same reasoning applies to the Schur complements S_h and S_N, i.e., the matrices obtained by eliminating the interior nodes of each Ω_i in a classical way; see [7]. Let u_N be Q-discrete (piecewise) harmonic if $a_Q(u_N, v_N) = 0$, for all i and all v_N belonging to $P^N_0(\Omega_i)$. The definition of h-discrete (piecewise) harmonic functions is analogous. It is easy to see that $\underline{u}^T S_N \underline{u} = a_Q(u_N, u_N)$ and that $\underline{u}^T S_h \underline{u} = a(u_h, u_h)$, where u_h and u_N are, respectively, Q and h-discrete harmonic and $\underline{u}$ is the vector of the nodal values on the interfaces of the substructures.

The matrices S_h and S_N are spectrally equivalent. Indeed, by the subassembly equation (10), it is enough to verify the spectral equivalence for each substructure separately. For the substructure Ω_i, we find:

$$\underline{u}^T S_N^{(i)} \underline{u} = a_{Q,\Omega_i}(u_N, u_N) \succeq a_{\Omega_i}(I_N^h(u_N), I_N^h(u_N)) \geq \tag{12}$$
$$a_{\Omega_i}(\mathcal{H}_h(I_N^h u_N), \mathcal{H}_h(I_N^h u_N)) = a_{\Omega_i}(u_h, u_h) = \underline{u}^T S_h^i \underline{u},$$

where I_N^h is the interpolation at the nodes of $\mathcal{T}_h$, $\mathcal{H}_h$ is the h-discrete harmonic extension of the interface values, and the subscript Ω_i indicates the restriction of the bilinear form to this substructure. Here, we have used FEM-SEM equivalence and the well-known minimizing property of the discrete harmonic extension. The reverse inequality is obtained in an analogous way.

In his Master's thesis [13], Pahl proposed the use of easily invertible finite element preconditioners B_h and $S_{h,WB}$ for K_h and S_h, respectively. If the condition number satisfies

$$\kappa(B_h^{-1} K_h) \leq C(N) \tag{13}$$

with a moderately increasing function $C(N)$, then a simple Rayleigh quotient argument shows that $\kappa(B_h^{-1} K_N) \preceq C(N)$, with an analogous bound for $S_{h,WB}$ and S_N. Since the evaluation of the action of ${B_h}^{-1}$ and $S_{h,WB}^{-1}$ is much cheaper, these are very efficient preconditioners.

Therefore, we only need to establish (13) and its analogue for S_h and $S_{h,WB}^{-1}$. We note that the triangulation $\mathcal{T}_h$ is nonregular, and that all the bounds of this form for Schwarz preconditioners established in the literature require some kind of inverse condition or regularity of the triangulation, which does not hold for the GLL mesh. In this paper we only analyze the iterative substructuring algorithms, but remark that the analysis for overlapping methods is a straightforward consequence of our techniques.

SOME ESTIMATES FOR NONREGULAR TRIANGULATIONS

We state here all the estimates necessary to extend the technical tools developed in [7] to the case of nonregular hexahedral triangulations. We let $\hat{K} = [-1, +1]^3$ be the reference element and K be its image under an affine mapping F. $K \subset \hat{\Omega}$ is an element of the triangulation $\mathcal{T}_{\hat{h}}$ with sides h_1, h_2 and h_3. The function u is a piecewise trilinear (Q_1) function defined in K. Notice that in this subsection we use hats to represent functions and points of $\hat{K}$.

The first result concerns the expressions of the L^2 and H^1 norms in terms of the nodal values. Let $\hat{e}_i$ be one of the coordinate directions of $\hat{K}$, and let $\hat{a}$, $\hat{b}$, $\hat{c}$ and $\hat{d}$ be the nodes on one of the faces that is perpendicular to $\hat{e}_i$, and let $\hat{a}'$, $\hat{b}'$, etc. be the corresponding points on the parallel face. The notation x_α denotes a generic node of K, and a, a', are the images of $\hat{a}$ and $\hat{a}'$, etc. The next lemma follows by changing variables, and by using the equivalence of any pair of norms in the finite dimensional space $Q_1(\hat{K})$.

Lemma 1.

$$\|u\|^2_{L^2(K)} \asymp h_1 h_2 h_3 \sum_{x_\alpha} (u(x_\alpha))^2 \tag{14}$$

$$\|\partial_{x_i} u\|^2_{L^2(K)} \asymp \frac{h_1 h_2 h_3}{h_i^2} \sum_{x_\alpha = a,b,c,d} (u(x_\alpha) - u(x'_\alpha))^2 \tag{15}$$

In the next lemma we give a bound on the gradient of a trilinear function in terms of bounds on the difference of the values at the nodes (vertices). The proof is elementary and is omitted.

Lemma 2. *Let u be trilinear in the element K such that $|u(a) - u(b)| \leq C|a - b|/r$ for some constants C and r, and for any two vertices a and b of the element K. Then*

$$|\nabla u| \leq \frac{C}{r}.$$

Lemma 3. *Let u be a trilinear function defined in K, and let ϑ be a C^1 function such that $|\nabla \vartheta| \leq C/r$ and $|\vartheta| \leq C$ for some constants C and r. Then*

$$|\partial_{x_i} I^{\hat{h}}(\vartheta u)|^2_{L^2(K)} \leq C(|u|^2_{H^1(K)} + r^{-2}||u||^2_{L^2(K)}). \tag{16}$$

Here C is independent of all the parameters, and $I^{\hat{h}}$ is the interpolation to a Q_1 function of the values in the vertices of K.

Proof. By equation (15), and letting h_1, h_2, and h_3 be the sides of the element K:

$$||\partial_{x_i} I^{\hat{h}}(\vartheta u)||^2_{L^2(\hat{K})} \preceq \frac{h_1 h_2 h_3}{h_i^2} \sum_{x=a,b,c,d} (u(x)\vartheta_{F^k}(x) - u(x')\vartheta_{F^k}(x'))^2.$$

Each term in the sum above can be bounded by

$$\begin{aligned}(u(x)\vartheta(x) - u(x)\vartheta(x') + u(x)\vartheta(x') - u(x')\vartheta(x'))^2 \leq \\ 2\left((u(x))^2(\vartheta(x) - \vartheta(x'))^2 + (u(x) - u(x'))^2(\vartheta(x'))^2\right).\end{aligned}$$

The bound on $\nabla\vartheta$ implies that $|\vartheta(x) - \vartheta(x')| \leq Ch_i/r$, and therefore

$$\begin{aligned}||\partial_{x_i} I^{\hat{h}}(\vartheta u)||^2_{L^2(\hat{K})} &\preceq \frac{h_1 h_2 h_3}{h_i^2}\Big(\sum_{x=a,b,c,d} (u(x) - u(x'))^2 + \sum_{x=a,b,c,d} (u(x))^2 \frac{h_i^2}{r^2}\Big) \\ &\preceq |u|^2_{H^1(\hat{K})} + r^{-2}||u||^2_{L^2(\hat{K})}, \end{aligned} \tag{17}$$

since ϑ is bounded.

□

TECHNICAL TOOLS

We introduce notations related to certain geometrical objects, since the iterative substructuring algorithms are based on subspaces directly related to the interiors of the substructures, the faces, edges and vertices. Let Ω_{ij} be the union of two substructures Ω_i and Ω_j, which share a common face, $\mathcal{F}_k$. Let $\mathcal{W}_j$ represent the wirebasket of the subdomain Ω_j, which is the union of all the edges and vertices of this subdomain. We note that a face in the interior of the region Ω is common to exactly two substructures, an interior edge is shared by more than two, and an interior vertex is common to still more substructures. All the substructures, faces, and edges are regarded as open sets.

The preconditioner $S_{h,WB}$ that we use is defined by subassembly of the matrices $S^{(i)}_{h,WB}$. Therefore we can restrict our analysis to one substructure. The results for the whole domain follow by a standard Rayleigh quotient argument. It is also enough to estimate the preconditioning of $\hat{S}_h$ by $\hat{S}_{h,WB}$, because these results can be translated into results for each substructure by the equivalences (5), (7), and (8).

The assumption that the $\{F_i\}_{i=1}^M$ are arbitrary smooth mappings improves the flexibility of the triangulation, but does not make the situation essentially different from the case of affine mappings. Therefore, without loss of generality, we assume, from now on, that the F_i are affine mappings.

In some of the following results, we state the result for substructures of diameter proportional to H, but prove the theorem only for a reference substructure. The introduction of the scaling factors into the final formulas is routine.

Lemma 4. *Let $\bar{u}^h_{\mathcal{W}_j}$ be the average value of u^h on $\mathcal{W}_j$, the wirebasket of the subdomain Ω_j. Then*

$$||u^h||^2_{L^2(W^j)} \leq C(1+\log(N))||u^h||^2_{H^1(\Omega_j)},$$

and

$$||u^h - \bar{u}^h_{\mathcal{W}_j}||^2_{L^2(W^j)} \leq C(1+\log(N))|u^h|^2_{H^1(\Omega_j)}.$$

Similar bounds also hold for an individual substructure edge.

Proof. In the reference substructure, we know that $P^{\hat{h}} \subset V^{\hat{h}}$, where $V^{\hat{h}}$ is a standard Q_1 finite element space defined on a shape regular triangulation that includes $\mathcal{T}^{\hat{h}}$. This can be done by refining appropriately all the elements of $\mathcal{T}^{\hat{h}}$ with sides larger than, for example, $3\hat{h}/2$.

Now we apply the well-known result for shape regular triangulations, lemma 4.3 in [7], to get both estimates, recalling that in the reference substructure $\hat{h} \asymp 1/N^2$.

□

In the abstract Schwarz convergence theory, the crucial point in the estimate of the rate of convergence of the algorithm is to demonstrate that all functions in the finite element space can be decomposed into components belonging to the subspaces in such a way that the sum of the resulting energies is uniformly, or almost uniformly, bounded with respect to the parameters H and N. The main technique for deriving

such a decomposition is the use of a suitable partition of unity. In the next two lemmas, we explicitly construct such a partition.

Lemma 5. *Let $\mathcal{F}_k$ be the common face between Ω_i and Ω_j, and let $\theta_{\mathcal{F}_k}$ be the function in $P^h(\Omega)$ that is equal to one at the interior nodes of $\mathcal{F}_k$, zero on the remainder of $(\partial\Omega_i \cup \partial\Omega_j)$, and discrete harmonic in Ω_i and Ω_j. Then*

$$|\theta_{\mathcal{F}_k}|^2_{H^1(\Omega_i)} \leq C(1+\log(N))H.$$

The same bound also holds for the other subregion Ω_j.

Proof. We define the functions $\hat{\theta}_{\mathcal{F}_k}$ and $\hat{\vartheta}_{\mathcal{F}_k}$ in the reference cube; $\theta_{\mathcal{F}_k}$ and $\vartheta_{\mathcal{F}_k}$ are obtained, as usual, by mapping. We construct a function $\hat{\vartheta}_{\mathcal{F}_k}$ having the same boundary values as $\hat{\theta}_{\mathcal{F}_k}$, and then prove the bound for the former. The standard energy minimizing property of discrete harmonic extensions then implies the bound for $\hat{\theta}_{\mathcal{F}_k}$. The six functions which correspond to the six faces of the cube also form a partition of unity at all nodes at the closure of the substructure except those on the wirebasket; this property is used in the next lemma.

We divide the substructure into twenty-four tetrahedra by connecting its center C to all the vertices and to all the six centers C_k of the faces, and by drawing the diagonals of the faces of $\hat{\Omega}$; see Fig 1.

The function $\hat{\vartheta}_{\mathcal{F}_k}$ associated with the face $\mathcal{F}_k$ is defined as being $1/6$ at the point C. The values at the centers of the faces are defined by $\hat{\vartheta}_{\mathcal{F}_k}(C_j) = \delta_{jk}$, where δ_{jk} is the Kronecker symbol. $\hat{\vartheta}_{\mathcal{F}_k}$ is defined to be linear on the segments CC_j for $j = 1, ..., 6$. The values inside each subtetrahedron defined by a segment CC_j and one edge of the cube are defined to be constant on the intersection of any plane through that edge and are given by the value, already known, at the segment CC_j. The values at the edge of the cube belonging to this subtetrahedron are then modified to be equal to zero. Next, the whole function $\hat{\vartheta}_{\mathcal{F}_k}$ is modified to be a piecewise Q_1 function by interpolating at the vertices of all the GLL nodes of the reference cube.

We claim that $|\nabla\hat{\vartheta}_{\mathcal{F}_k}(x)| \leq C/r$, where x is a point belonging to any element K that does not touch any edge of the cube, and r is the distance between the center of K and the closest edge of the cube. Let $\overline{ab}$ be a side of K. We analyze in detail the situation depicted in Fig. 2, where $\overline{ab}$ is parallel to CC_k. Let e be the intersection of the plane containing these two segments with the edge of the cube that is closest to $\overline{ab}$. Then $|\hat{\vartheta}_{\mathcal{F}_k}(b) - \hat{\vartheta}_{\mathcal{F}_k}(a)| \preceq D$, by construction of $\hat{\vartheta}_{\mathcal{F}_k}$, where D is the size of the radial projection of $\overline{ab}$ on CC_k. By similarity of triangles, we may write:

$$|\hat{\vartheta}_{\mathcal{F}_k}(b) - \hat{\vartheta}_{\mathcal{F}_k}(a)| \leq C\frac{\text{dist}(a,b)}{r'}, \tag{18}$$

where r' is the distance between e and the midpoint of $\overline{ab}$. Here we have used that the distance between e and CC_k is of order 1. If the segment $\overline{ab}$ is not parallel to CC_k, the difference $|\hat{\vartheta}_{\mathcal{F}_k}(b) - \hat{\vartheta}_{\mathcal{F}_k}(a)|$ is even smaller, and (18) is still valid. Notice that r' is within a multiple of 2 of r. Therefore Lemma 2 implies that $|\nabla\hat{\vartheta}_{\mathcal{F}_k}(x)| \leq C/r$.

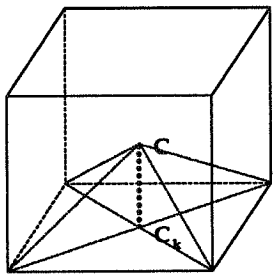


Figure 1: One of the segments CC_k

In order to estimate the energy of $\hat{\vartheta}_{\mathcal{F}_k}$, we start with the elements K that touch one of the edges of the face $\mathcal{F}_k$. Let h_3 be the largest side of one of these elements. Since the nodal values of $\hat{\vartheta}_{\mathcal{F}_k}$ at K are 0, 1, and 1/6,

$$|\hat{\vartheta}_{\mathcal{F}_k}|_{H^1(K)} \leq Ch_3,$$

by a simple use of equation (15). By summing over K, we conclude that the energy of $\hat{\vartheta}_{\mathcal{F}_k}$ is bounded independently of N for the union of all elements that touch one of the edges of the face $\mathcal{F}_k$.

To estimate the contribution to the energy from the rest of the substructure, we consider one subtetrahedron at a time and introduce cylindrical coordinates using the substructure edge, that belongs to the subtetrahedron, as the z-axis. The bound now follows from the bound on the gradient given above and from elementary considerations. We refer to [7] for more details.

□

The following lemma corresponds to Lemma 4.5 in [7]. This lemma and the previous one are the keys for avoiding $H_{00}^{1/2}$ estimates and extension theorems.

Lemma 6. *Let $\vartheta_{\mathcal{F}_k}(x)$ be the function introduced in the proof of Lemma 5, let $\mathcal{F}_k$ be a face of the substructure Ω_j, and let I^h denote the interpolation operator associated with the finite element space P^h and the image of the GLL points under the mapping F_j. Then,*

$$\sum_k I^h(\vartheta_{\mathcal{F}_k} u)(x) = u(x),$$

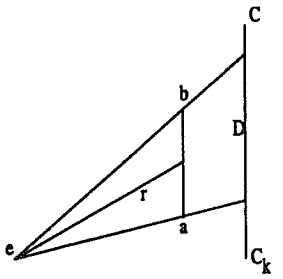


Figure 2: Geometry underlying equation (18)

for all nodal points $x \in \overline{\Omega}_j$ that do not belong to the wirebasket $\mathcal{W}_j$, and

$$|I^h(\vartheta_{\mathcal{F}_k} u)|^2_{H^1(\Omega_j)} \leq C(1+\log(N))^2 ||u||^2_{H^1(\Omega_j)}.$$

Proof. The first part is trivial from the construction of $\hat{\vartheta}_{\mathcal{F}_k}$ made in the previous lemma. For the second part, we first estimate the sum of the energy of all the elements K that touch the wirebasket. The nodal values of the interpolator $I^{\hat{h}}(\hat{\vartheta}_{\mathcal{F}_k}\hat{u}^{\hat{h}})$ in such an element are 0,0,0,0, $\hat{u}(a)$, $\hat{u}(b)$, $\hat{\vartheta}_{\mathcal{F}_k}(c)\hat{u}(c)$ and $\hat{\vartheta}_{\mathcal{F}_k}(d)\hat{u}(d)$; $\hat{\vartheta}_{\mathcal{F}_k}$ lies between 0 and 1. Moreover, we denote by h_3 the side of K that is larger than the other two sides h_1 and $h_2 = h_1$. Note that this larger side is parallel to the closest wirebasket edge. Since $h_1 \leq h_3$, and using equation (15), we obtain:

$$|I^{\hat{h}}(\hat{\vartheta}_{\mathcal{F}_k}\hat{u})|^2_{H^1(K)} \leq Ch_3(\hat{u}^2(a) + \hat{u}^2(b) + (\hat{\vartheta}_{\mathcal{F}_k}(c)\hat{u}(c))^2 + (\hat{\vartheta}_{\mathcal{F}_k}(d)\hat{u}(d))^2).$$

Then, by using the expression of the L^2-norm in the two segments that are parallel to the edge, and lemma 4, we have:

$$\sum_K |I^{\hat{h}}(\hat{\vartheta}_{\mathcal{F}_k}\hat{u})|^2_{H^1(K)} \leq C(1+\log(N))||\hat{u}||^2_{H^1(\Omega_j)},$$

where the sum is taken over all elements K that touch the boundary of the face $\mathcal{F}_k$.

We next bound the energy of the interpolant for the other elements. Since $|\nabla\hat{\vartheta}_{\mathcal{F}_k}| \leq C/r$, where r is the distance between the element K and the nearest edge of $\hat{\Omega}$ (see the proof of the previous lemma), Lemma 3 implies that

$$\sum_{K\subset\hat{\Omega}} |I^{\hat{h}}(\hat{\vartheta}_{\mathcal{F}_k}\hat{u})|^2_{H^1(K)} \leq C \sum_{K\subset\hat{\Omega}} (|\hat{u}|_{H^1(K)} + r^{-2}||\hat{u}||^2_{L^2(K)}),$$

where the sum is taken over all elements K that do not touch the edges of $\hat{\Omega}$.

The bound of the first term in the sum is trivial. To bound the second term, we partition the elements of $\hat{\Omega}$ into groups, in accordance with the closest edge of $\hat{\Omega}$; the exact rule for the assignment of the elements that are halfway between is of no importance. For each edge of the wirebasket, we use a local cylindrical coordinate system with the z axis coinciding with the edge, and the radial direction, r, normal to the edge. In cylindrical coordinates, we estimate the sum by an integral

$$\sum_{K\subset\hat{\Omega}} r^{-2}||\hat{u}||^2_{L^2(K)} \leq C\int_{r=\hat{h}}^{C}\int_\theta\int_z (\hat{u})^2\frac{r}{r^2}drd\theta dz.$$

The integral with respect to z can be bounded by using Lemma 4. We obtain

$$\sum_{K\subset\hat{\Omega}} r^{-2}||\hat{u}||^2_{L^2(K)} \leq C(1+\log(C/\hat{h}))||\hat{u}||^2_{H^1(\hat{\Omega})}\int_{r=\hat{h}}^{C} r^{-1}dr$$

and thus

$$\sum_{K\subset\hat{\Omega}} |I^{\hat{h}}(\hat{\vartheta}_{\mathcal{F}_k}\hat{u})|^2_{H^1(K)} \leq C(1+\log(C/\hat{h}))^2 ||\hat{u}||^2_{H^1(\hat{\Omega})}.$$

□

Lemma 7. *Let $\bar{u}^h_{\partial\mathcal{F}_k}$, and $\bar{u}^h_{W^k}$ be the averages of u^h on $\partial\mathcal{F}_k$, and W^k, respectively. Then,*

$$(\bar{u}^h_{\partial\mathcal{F}_k})^2 \leq C\frac{1}{H}||u^h||^2_{L^2(\partial\mathcal{F}_k)},$$

$$(\bar{u}^h_{W^k})^2 \leq C\frac{1}{H}||u^h||^2_{L^2(W^k)}.$$

The proofs are direct consequences of the Cauchy–Schwarz inequality.

Lemma 8. *Let u^h be zero on the mesh points of the faces of Ω_j and discrete harmonic in Ω_j. Then*

$$|u^h|^2_{H^1(\Omega_j)} \leq C||u^h||^2_{L^2(W^j)}.$$

This result follows by estimating the energy norm of the zero extension of the boundary values by means of equation (15) and by noting that the harmonic extension has a smaller energy.

ITERATIVE SUBSTRUCTURING ALGORITHMS

The first algorithm we analyze is a wirebasket based method, based on Algorithm 6.4 in [7]. This is a block-diagonal preconditioner after transforming the original matrix to a convenient basis.

To use the abstract framework of Schwarz methods [7], we only need to prescribe spaces whose union is the whole space, and the corresponding bilinear forms.

Each internal face $\mathcal{F}_k$ generates a local space $V_{\mathcal{F}_k}$ of all the h-discrete harmonic functions that are zero at all the interface nodes that do not belong to this face. Notice that the functions belonging to $V_{\mathcal{F}_k}$ have support in the union of the two substructures $\overline{\Omega_i}$ and $\overline{\Omega_j}$ that share the face $\mathcal{F}_k$. The bilinear form used for this space is $a(\cdot,\cdot)$.

We also define a wirebasket subspace that is the range of the following interpolation operator:

$$I^h_W u^h = \sum_{x_k\in\mathcal{W}_h} u^h(x_k)\varphi_k + \sum_k \bar{u}^h_{\partial F^k}\theta_{F^k}.$$

Here, φ_k is the discrete harmonic extension of the standard nodal basis functions ϕ_k, $\mathcal{W}_h$ is the set of nodes in the union of all the wirebaskets, and $\bar{u}^h_{\partial F^k}$ is the average of u^h on ∂F^k. The bilinear form for this coarse subspace is given by

$$b_0(u,u) = (1+\log(N))\sum_i k_i \inf_{c_i} ||u - c_i||^2_{L^2(\mathcal{W}_i)}.$$

These subspaces and bilinear forms define, via the Schwarz framework, a preconditioner of S_h that we call $S_{h,WB}$.

Theorem 1. *For the preconditioner $S_{h,WB}$, we have*

$$\kappa(S_{h,WB}^{-1} S_N) \leq C(1 + log(N))^2,$$

where the constant C is independent of the N, H, and the values k_i of the coefficient.

Proof. We apply, word by word, the proof of theorem 6.4 in [7] to the matrix S_h, using the tools developed in the previous section. This gives

$$\kappa(S_{h,WB}^{-1} S_h) \leq C(1 + log(N))^2.$$

The harmonic FEM-SEM equivalence (12) and a Rayleigh quotient argument complete the proof.

□

We do not give the complete proof here because it would be a mere restatement of the proof in [7].

The next algorithm is obtained from the previous one by the discrete harmonic FEM-SEM equivalence, by which we find a preconditioner $S_{N,WB}$ from the preconditioner $S_{h,WB}$ studied above. Each face subspace, related to a face $\mathcal{F}_k$, is composed of the set of all Q-discrete harmonic functions that are zero at all the interface nodes that do not belong to the interior of the face $\mathcal{F}_k$.

The wirebasket subspaces are defined as before, by prescribing the values at the GLL nodes on a face to be equal to the average of the function on the boundary of the face. The bilinear forms used for the face and wirebasket subspaces are $a_Q(\cdot,\cdot)$ and $b_0(\cdot,\cdot)$, respectively. Notice that this is the wirebasket method based on GLL quadrature given in [19].

The following lemma shows the equivalence of the two functions u_N and u_h with respect to the bilinear form $b_0(\cdot,\cdot)$.

Lemma 9. *Let u_h be a Q_1 finite element function on the GLL mesh of the interval $I = [-1,+1]$, and let u_N be its polynomial interpolant. Then*

$$\inf_c ||u_h - c||^2_{L^2(I)} \asymp \inf_c ||u_N - c||^2_{L^2(I)}$$

Proof. We prove only the $\leq$ part. The inequality without the infimum is valid for the constant c_r that realizes the inf in the right hand side by the FEM-SEM equivalence. By taking the inf in the left hand side we preserve the inequality.

□

Theorem 2. *For the preconditioner $S_{N,WB}$, we have*

$$\kappa(S_{N,WB}^{-1} S_N) \leq C(1 + \log(N))^2$$

where the constant is independent of the parameters H, N and the values k_i of the coefficient. *Proof.* In this proof, the functions with indices h and N are all discrete harmonic functions with respect to the appropriate norms, related in the same way as u_N and u_h, i.e. $u_h = \mathcal{H}_h(I_N^h u_N)$. According to equation (10), it is enough to analyze one substructure Ω_i at a time, and prove the following equivalence:

$$b_{0,\mathcal{W}_i}(u_N, u_N) + \sum_{\mathcal{F}_k \subset \Omega_i} |u_N - \overline{u}_{N,\partial\mathcal{F}_k} \theta_{N,\mathcal{F}_k}|^2_{H^1(\Omega_i)} \asymp \tag{19}$$
$$b_{0,\mathcal{W}_i}(u_h, u_h) + \sum_{\mathcal{F}_k \subset \Omega_i} |u_h - \overline{u}_{h,\partial\mathcal{F}_k} \theta_{h,\mathcal{F}_k}|^2_{H^1(\Omega_i)}.$$

We prove only the $\leq$ part; the proof of the other inequality is analogous. Lemma 9 gives an upper bound of the first term of the left hand side by the corresponding term in the right hand side.

Each term in the sum on the left hand side can be bounded by

$$2|u_N - \overline{u}_{h,\partial\mathcal{F}_k} \theta_{N,\mathcal{F}_k}|^2_{H^1(\Omega_i)} + 2|(\overline{u}_{h,\partial\mathcal{F}_k} - \overline{u}_{N,\partial\mathcal{F}_k}) \theta_{N,\mathcal{F}_k}|^2_{H^1(\Omega_i)}.$$

The first term of this expression can be bounded by the corresponding term on the right hand side by interpolation and the harmonic FEM-SEM equivalence. The second term is bounded by

$$H(1 + \log(N))|\overline{u}_{h,\partial\mathcal{F}_k} - \overline{u}_{N,\partial\mathcal{F}_k}|^2 =$$
$$H(1 + \log(N))|\overline{(u - c_{h,\mathcal{W}_i})}_{h,\partial\mathcal{F}_k} - \overline{(u - c_{h,\mathcal{W}_i})}_{N,\partial\mathcal{F}_k}|^2,$$

where $c_{h,\mathcal{W}_i}$ is the average of u_h over $\mathcal{W}_i$. Here we have used the estimate on the energy norm of $\theta_{h,\mathcal{F}_k}$ which implies a similar estimate for $\theta_{N,\mathcal{F}_k}$. Applying the Cauchy-Schwarz inequality, as in lemma 7, and the FEM-SEM equivalence, we can bound this last expression in terms of the first term in the right hand side of equation (19).

□

REFERENCES

[1] C. Bernardi and Y. Maday. Polynomial interpolation results in sobolev spaces. *J. Comput. Appl. Math.*, 43:53 - 80, 1992.

[2] Christine Bernardi and Yvon Maday. *Approximations Spectrales de Problèmes aux Limites Elliptiques*, volume 10 of *Mathèmatiques & Applications*. Springer-Verlag France, Paris, 1992.

[3] James H. Bramble, Joseph E. Pasciak, and Alfred H. Schatz. The construction of preconditioners for elliptic problems by substructuring, IV. *Math. Comp.*, 53:1–24, 1989.

[4] Claudio Canuto. Stabilization of spectral methods by finite element bubble functions. *Comp. Methods Appl. Mech. Eng.*, 116(1–4):13–26, 1994.

[5] Tony F. Chan, David E. Keyes, Gérard A. Meurant, Jeffrey S. Scroggs, and Robert G. Voigt, editors. *Fifth International Symposium on Domain Decomposition Methods for Partial Differential Equations*, Philadelphia, PA, 1992. SIAM. Held in Norfolk, VA, May 6–8, 1991.

[6] M. O. Deville and E. H. Mund. Finite-element preconditioning for pseudospectral solutions of elliptic problems. *Siam J. Sci. Stat. Comput.*, 11(2):311 - 342, March 1990.

[7] Maksymilian Dryja, Barry F. Smith, and Olof B. Widlund. Schwarz analysis of iterative substructuring algorithms for elliptic problems in three dimensions. *SIAM J. Numer. Anal.*, 31(6), December 1994.

[8] Maksymilian Dryja and Olof B. Widlund. Additive Schwarz methods for elliptic finite element problems in three dimensions. In Tony F. Chan, David E. Keyes, Gérard A. Meurant, Jeffrey S. Scroggs, and Robert G. Voigt, editors, *Fifth International Symposium on Domain Decomposition Methods for Partial Differential Equations*, Philadelphia, PA, 1992. SIAM.

[9] Paul F. Fischer. Parallel Domain Decomposition for Incompressible Fluid Dynamics. In Alfio Quarteroni, Yuri A. Kuznetsov, Jacques Périaux, and Olof B. Widlund, editors, *Domain Decomposition Methods in Science and Engineering: The Sixth International Conference on Domain Decomposition*, volume 157. AMS, 1994. Held in Como, Italy, June 15–19, 1992.

[10] David E. Keyes and Jinchao Xu, editors. *Domain Decomposition Methods in Science and Engineering*, Providence, R.I., 1995. AMS. Proceedings of the Seventh International Conference on Domain Decomposition, October 27-30, 1993, The Pennsylvania State University.

[11] Yvon Maday, Dan Meiron, Anthony T. Patera, and Einar M. Rønquist. Analysis of iterative methods for the steady and unsteady Stokes problem: Application of spectral element discretization. *SIAM J. Sci. Comp.*, 14(2):310–337, 1993.

[12] Yvon Maday and Anthony T. Patera. Spectral element methods for the Navier-Stokes equations. In A.K. Noor and J.T. Oden, editors, *State of the Art Surveys in Computational Mechanics*, New York, 1989. ASME.

[13] Shannon S. Pahl. Schwarz type domain decomposition methods for spectral element discretizations. Master's thesis, Department of Computational and Applied Mathematics, University of the Witwatersrand, Johannesburg, South Africa, December 1993.

[14] Seymour V. Parter and Ernest E. Rothman. Preconditioning legendre spectral collocation approximation to elliptic problems. *SIAM J. Numer. Anal.*, 32(2), April 1995.

[15] Luca F. Pavarino. *Domain Decomposition Algorithms for the p-version Finite Element Method for Elliptic Problems.* PhD thesis, Courant Institute, New York University, September 1992.

[16] Luca F. Pavarino. Additive Schwarz methods for the p-version finite element method. *Numer. Math.*, 66(4):493–515, 1994.

[17] Luca F. Pavarino. Some Schwarz algorithms for the p-version finite element method. In Alfio Quarteroni, Yuri A. Kuznetsov, Jacques Périaux, and Olof B. Widlund, editors, *Domain Decomposition Methods in Science and Engineering: The Sixth International Conference on Domain Decomposition*, volume 157. AMS, 1994. Held in Como, Italy, June 15–19, 1992.

[18] Luca F. Pavarino and Olof B. Widlund. Iterative substructuring methods for spectral elements: Problems in three dimensions based on numerical quadrature. Technical Report 663, Courant Institute of Mathematical Sciences, Department of Computer Science, May 1994.

[19] Luca F. Pavarino and Olof B. Widlund. A polylogarithmic bound for an iterative substructuring method for spectral elements in three dimensions. Technical Report 661, Courant Institute of Mathematical Sciences, Department of Computer Science, March 1994. To appear in *SIAM J. Numer. Anal.*

[20] A. Quarteroni and E. Zampieri. Finite element preconditioning for legendre spectral collocation approximations to elliptic equations and systems. *SIAM J. Numer. Anal.*, 29:917 - 936, 1992.

[21] Alfio Quarteroni, Yuri A. Kuznetsov, Jacques Périaux, and Olof B. Widlund, editors. *Domain Decomposition Methods in Science and Engineering: The Sixth International Conference on Domain Decomposition*, volume 157. AMS, 1994. Held in Como, Italy, June 15–19, 1992.

RECENT DEVELOPMENT OF MULTIGRID ALGORITHMS FOR MIXED AND NONCONFORMING METHODS FOR SECOND ORDER ELLIPTIC PROBLEMS

Zhangxin Chen
Department of Mathematics
Box 156, Dedman College
Southern Methodist University
Dallas, TX 75275-0156

Richard E. Ewing
Institute for Scientific Computation
and Department of Mathematics
Texas A&M University
College Station, TX 77843-3404

Abstract

Multigrid algorithms for nonconforming and mixed finite element methods for second order elliptic problems on triangular and rectangular finite elements are considered. The construction of several coarse-to-fine intergrid transfer operators for nonconforming multigrid algorithms is discussed. The equivalence between the nonconforming and mixed finite element methods with and without projection of the coefficient of the differential problems into finite element spaces is described.

INTRODUCTION

In this paper we consider multigrid algorithms for numerical solution of the model problem

$$\begin{aligned} -\nabla \cdot (a \nabla u) &= f \quad \text{in } \Omega, \\ u &= 0 \quad \text{on } \partial\Omega, \end{aligned} \tag{1.1}$$

using nonconforming and mixed finite element methods, where $\Omega \subset \mathbb{R}^n$, $n = 2, 3$ is a simply connected bounded polygonal domain with the boundary $\partial\Omega$, $f \in L^2(\Omega)$, and the coefficient $a \in L^\infty(\Omega)$ satisfies

$$0 < a_1 \leq a(x) \leq a_2, \qquad x \in \Omega, \tag{1.2}$$

with fixed constants a_1, a_2. The $\mathcal{W}$-cycle multigrid algorithm for numerically solving (1.1) using the P_1-nonconforming finite element method over triangles has been extensively studied in [6, 9, 15]. It has been shown that the $\mathcal{W}$-cycle algorithm with a particular coarse-to-fine intergrid transfer operator (the so-called averaging operator) is convergent under the assumption that the number of smoothing iterations on all levels is big enough. In [18] a convergence analysis for multigrid algorithms for (1.1) on triangular and rectangular elements, based on the abstract theory in [8] for multigrid methods with nonnested spaces, has been introduced. This analysis applies to both the $\mathcal{W}$-cycle and the variable $\mathcal{V}$-cycle. It was shown in [18] that optimal convergence properties of the $\mathcal{W}$-cycle multigrid algorithm and uniform condition number estimates for the variable $\mathcal{V}$-cycle preconditioner can be established with the averaging intergrid transfer operator.

In this paper the $\mathcal{V}$-cycle and $\mathcal{W}$-cycle multigrid algorithms for numerically solving (1.1) using the nonconforming finite element method both over triangles and rectangles are considered in detail. Special attention is paid to the construction of several coarse-to-fine intergrid transfer operators for the nonconforming multigrid algorithms. In particular, we introduce a new intergrid transfer operator and indicate the convergence of the $\mathcal{V}$-cycle algorithm, which has not been proved before. Our preliminary results show that a similar operator also works for the biharmonic problem.

The multigrid algorithms for mixed finite element methods are also considered here. The mixed methods require the solution of linear systems in the form of a saddle point problem, which can be expensive to solve. An alternate approach was suggested by means of a nonmixed formulation. Namely, it has been shown that the mixed methods are equivalent to a modification of nonconforming Galerkin methods [1, 2, 14, 27]. The modified nonconforming methods yield a symmetric and positive definite problem (i.e., a minimization problem). However, various bubble functions have been used to establish the equivalence between the two methods, which can be again expensive from the computational point of view. In [15] a new approach has been introduced to establish the equivalence between the mixed and nonconforming methods without using the bubble functions. The projection of the coefficient a of the differential equation (1.1) into finite element spaces has been incorporated into the mixed formulation. Recently, we have been asked if the equivalence still holds without the coefficient projection. A positive answer will be given in this paper. In particular, a comparison between the usual and projected mixed methods is given, and we show that the latter version gives us considerable computational savings, without any loss of accuracy, as observed before [20].

The remainder of the paper is organized as follows. In the next section multigrid algorithms for the P_1-nonconforming method over triangles are developed. Then in the third section multigrid algorithms for triangular mixed methods are considered.

An extension to the corresponding rectangular elements is carried out in the fourth section. Finally, numerical experiments on the performance of the present approaches are given in the fifth section. The later analysis is carried out for the two-dimensional case; it works for the three-dimensional case without substantial changes [15, 17, 16].

NONCONFORMING MULTIGRID ALGORITHMS

Problem (1.1) is recast in weak form as follows. We define the bilinear form $a(\cdot,\cdot)$ as follows:

$$a(v,w) = (a\nabla v, \nabla w), \qquad v, w \in H^1(\Omega),$$

where $(\cdot,\cdot)$ denotes the $L^2(\Omega)$ or $(L^2(\Omega))^2$ inner product, as appropriate. Then the weak form of (1.1) for the solution $u \in H_0^1(\Omega)$ is

$$a(u,v) = (f,v), \qquad \forall\, v \in H_0^1(\Omega). \tag{2.1}$$

For $0 < h < 1$, let $\mathcal{E}_h$ be a triangulation of Ω into triangles of size h and define the P_1-nonconforming finite element space

$$V_h = \{v \in L^2(\Omega) : \ v|_E \text{ is linear for all } E \in \mathcal{E}_h, v \text{ is continuous at the midpoints of interior edges and vanishes at the midpoints of edges on } \partial\Omega\}.$$

Associated with V_h, we introduce a bilinear form on $V_h \oplus H_0^1(\Omega)$ by

$$a_h(v,w) = \sum_{E\in\mathcal{E}_h} (a\nabla v, \nabla w)_E, \qquad v, w \in V_h \oplus H_0^1(\Omega),$$

where $(\cdot,\cdot)_E$ is the $L^2(E)$ inner product. Then the P_1-nonconforming finite element discretization of (1.1) is to find $u_h \in V_h$ such that

$$a_h(u_h, v) = (f,v), \qquad \forall\, v \in V_h. \tag{2.2}$$

After we use a set of bases in V_h, (2.2) leads to the following linear system:

$$A_h u_h = F_h, \tag{2.3}$$

where A_h is symmetric and positive definite.

To develop a multigrid algorithm for (2.1), we need to assume a structure to our family of partitions. Let h_0 and $\mathcal{E}_{h_0} = \mathcal{E}_0$ be given. For each integer $1 \le k \le K$, let $h_k = 2^{-k}h_0$ and $\mathcal{E}_{h_k} = \mathcal{E}_k$ be constructed by connecting the midpoints of the edges of the triangle in $\mathcal{E}_{k-1}$, and let $\mathcal{E}_h = \mathcal{E}_K$ be the finest grid. We replace subscript h_k simply by subscript k.

Let $I_{k-1}^k : V_{k-1} \to V_k$ denote some as yet unspecified coarse-to-fine intergrid transfer operator. By an abuse of notation, we also denote by I_{k-1}^k the matrix of this operator with respect to the bases $\{\psi_1^{k-1}, \ldots, \psi_{m_{k-1}}^{k-1}\}$ of V_{k-1} and $\{\psi_1^k, \ldots, \psi_{m_k}^k\}$ of

V_k, and $I_k^{k-1} : V_k \to V_{k-1}$ the transpose of I_{k-1}^k. Finally, let ω_k indicate a parameter, which is chosen to be not smaller than the largest eigenvalue of A_k.

We now formulate our multigrid algorithm for (2.3). The following algorithm defines a multigrid operator $B_k : V_k \to V_k$.

MULTIGRID ALGORITHM 2.1. Let $1 \leq k \leq K$, and μ be a positive integer. Set $B_0 = A_0^{-1}$. Assume that B_{k-1} has been defined and define $B_k g$ for $g \in V_k$ as follows:

1. Set $x^0 = 0$ and $q^0 = 0$.

2. Define x^l for $l = 1, \ldots, m(k)$ by

$$x^l = x^{l-1} + \omega_k^{-1}(g - A_k x^{l-1}).$$

3. Define $y^{m(k)} = x^{m(k)} + I_{k-1}^k q^p$, where q^i for $i = 1, \ldots, \mu$ is defined by

$$q^i = q^{i-1} + B_{k-1}\left[I_k^{k-1}\left(g - A_k x^{m(k)}\right) - A_{k-1} q^{i-1}\right]. \tag{2.4}$$

4. Define y^l for $l = m(k)+1, \ldots, 2m(k)$ by

$$y^l = y^{l-1} + \omega_k^{-1}\left(g - A_k y^{l-1}\right).$$

5. Set $B_k g = y^{2m(k)}$.

In Algorithm 2.1, $m(k)$ gives the number of smoothing iterations and can vary as a function of k. If $\mu = 1$, we have a $\mathcal{V}$-cycle multigrid algorithm. If $\mu = 2$, we have a $\mathcal{W}$-cycle algorithm. A variable $\mathcal{V}$-cycle algorithm is one in which the number of smoothings $m(k)$ increase exponentially as k decreases (i.e., $\mu = 1$ and $m(k) = 2^{K-k}$).

We now consider the problem of how to construct a coarse-to-fine intergrid transfer operator I_{k-1}^k. We first review three known operators, and then introduce a new one.

EXAMPLE 1. The first operator is the so-called averaging operator, which was first defined in [6] and [9]. For $v \in V_{k-1}$, let q be a midpoint of an edge of a triangle in $\mathcal{E}_k$; then we define $I_k v$ by

$$\left(I_{k-1}^k v\right)(q) = \begin{cases} 0 & \text{if } q \in \partial\Omega, \\ v(q) & \text{if } q \notin \partial E \text{ for any } E \in \mathcal{E}_{k-1}, \\ \frac{1}{2}\{v|_{E_1}(q) + v|_{E_2}(q)\} & \text{if } q \in \partial E_1 \cap \partial E_2 \text{ for some } E_1, E_2 \in \mathcal{E}_{k-1}. \end{cases}$$

With this operator, as mentioned in the introduction, it has been first shown in [6, 9] that the $\mathcal{W}$-cycle algorithm (i.e., $\mu = 2$) is convergent under the assumption that the number of smoothing iterations on all levels is big enough (following the standard proof of convergence for conforming methods [4, 3]). Then in [18] a convergence analysis for Algorithm 2.1 was given, which establishes optimal convergence properties of the $\mathcal{W}$-cycle multigrid algorithm and uniform condition number estimates for the variable $\mathcal{V}$-cycle preconditioner; see the theorem below. Since this operator does not

preserve the energy norm, the standard proof of convergence given in [5, 8] for the conforming finite elements does not work for the nonconforming $\mathcal{V}$-cycle. In fact, while we can establish the stability property [6, 9, 15]

$$a_k(I_k v, I_k v) \leq C a_{k-1}(v, v), \qquad \forall\, v \in V_{k-1}, \tag{2.5}$$

with C independent on k, the constant C is in general bigger than two, as observed in [18].

EXAMPLE 2. The second example was originally described in [33], and then used in [17] for analyzing domain decomposition methods for mixed finite element methods. If $v \in V_{k-1}$ and $E \in \mathcal{E}_{k-1}$ with the vertices q_i and the midpoints $\overline{q}_i$ of its edges, $i = 1, 2, 3$, then

$$\begin{aligned}
&I_{k-1}^k v(\overline{q}_i) = v(\overline{q}_i), && i = 1, 2, 3, \\
&I_{k-1}^k v(q_i) = \tfrac{1}{\mathcal{N}_1} \textstyle\sum_j v(\overline{q}'_j) && \text{if } q_i \notin \partial\Omega, \\
&I_{k-1}^k v(q_i) = \tfrac{1}{\mathcal{N}_2} \textstyle\sum_j v(\overline{q}''_j) && \text{if } q_i \in \partial\Omega,
\end{aligned}$$

where $\mathcal{N}_1$ and $\mathcal{N}_2$ are the number of the adjacent midpoints $\overline{q}'_j$ and $\overline{q}''_j$ to q_i of the interior edges and the boundary edges of the elements in $\mathcal{E}_{k-1}$, respectively.

EXAMPLE 3. The third example [17, 21] is very similar to that in Example 2. If $v \in V_{k-1}$ and $E \in \mathcal{E}_{k-1}$ with the vertices q_i and the midpoints $\overline{q}_i$ of its edges, $i = 1, 2, 3$, then

$$\begin{aligned}
&I_{k-1}^k v(\overline{q}_i) = v(\overline{q}_i), && i = 1, 2, 3, \\
&I_{k-1}^k v(q_i) = \tfrac{1}{\mathcal{N}_1} \textstyle\sum_{q_i \in K_j} v|_{K_j}(q_i) && \text{if } q_i \notin \partial\Omega, \\
&I_{k-1}^k v(q_i) = \tfrac{1}{\mathcal{N}_2} \textstyle\sum_j v(\overline{q}''_j) && \text{if } q_i \in \partial\Omega,
\end{aligned}$$

where $\mathcal{N}_1$ is the number of elements $K_j \in \mathcal{E}_{k-1}$ that meet at q_i and $\mathcal{N}_2$ is defined as in Example 2.

Note that Examples 2 and 3 define the value of $I_{k-1}^k v$ at the vertices of elements in $\mathcal{E}_k$ and thus lead to a continuous piecewise linear function on $\mathcal{E}_k$. Hence $I_{k-1}^k v$ is obviously in V_k. Also, since the operators in Examples 2 and 3 do not preserve the energy norm, we can only establish the optimal convergence properties of the $\mathcal{W}$-cycle multigrid algorithm and the uniform condition number estimates for the variable $\mathcal{V}$-cycle preconditioner via the standard convergence proof [4, 8], as in Example 1.

With the three definitions above, we now state a convergence result, whose proof is given in [18].

The convergence rate for Algorithm 2.1 on the kth level is measured by a convergence factor δ_k that satisfy

$$|a_k((I - B_k A_k)v, v)| \leq \delta_k a_k(v, v), \qquad \forall\, v \in V_k.$$

Theorem. (i) *Define B_k by $\mu = 2$ for all k in Algorithm 2.1. Then there exists $C > 0$ independent of k such that for a large enough m*

$$\delta_k \leq \delta \equiv \frac{C}{C + \sqrt{m}}.$$

(ii) *Define B_k by $\mu = 1$ and $m(k) = 2^{K-k}$ for $k = 1, \ldots, K$ in Algorithm 2.1. Then there are η_0, $\eta_1 > 0$, independent of k, such that*

$$\eta_0 a_k(v, v) \leq a_k(B_k A_k v, v) \leq \eta_1 a_k(v, v), \qquad \forall v \in V_k,$$

with $\eta_0 \geq \frac{m(k)^{1/2}}{C+m(k)^{1/2}}$ and $\eta_1 \leq \frac{C+m(k)^{1/2}}{m(k)^{1/2}}$.

EXAMPLE 4. We now define the operator $I_{k-1}^k : V_{k-1} \to V_k$ by

$$a_k(I_{k-1}^k v, w) = a_k(v, w), \quad \forall v \in V_{k-1}, w \in V_k. \tag{2.6}$$

With this definition, the inequality (2.5) is trivially satisfied with $C = 1$. Hence the abstract theory in [8] can be applied to show convergence of both the $\mathcal{V}$-cycle and the $\mathcal{W}$-cycle algorithms with one smoothing iteration. However, the I_{k-1}^k in (2.6) is not practical. The cost to obtain $I_{k-1}^k v$ for $v \in V_{k-1}$ is almost the same as that to solve the original linear system. The problem is that $I_{k-1}^k v$ cannot be explicitly determined. To get around this obstacle, we now consider the operator $I_k^{k-1} : V_k \to V_{k-1}$ defined by

$$a_{k-1}(I_k^{k-1} v, w) = a_k(v, I_{k-1}^k w), \quad \forall v \in V_k, w \in V_{k-1}. \tag{2.7}$$

The operator I_k^{k-1} can be explicitly determined by the simple relation (the proof will be presented in a forthcoming paper)

$$(I_k^{k-1} v)(q_1) = \frac{1}{2}(v(q_A) + v(q_B)),$$

for $v \in V_k$ (see Figure 1). With use of the operator I_k^{k-1} and its transpose I_{k-1}^k in Algorithm 2.1, we can prove convergence of both the $\mathcal{V}$-cycle and the $\mathcal{W}$-cycle algorithms. This will be given in the forthcoming paper. We remark that the same construction of the operator I_k^{k-1} can be carried out for Morley's elements for the biharmonic problem.

MULTIGRID ALGORITHMS FOR MIXED METHODS

The Raviart-Thomas space [31] over triangles is given by

$$\begin{aligned}
\Lambda_h &= \{v \in (L^2(\Omega))^2 : v|_E = (a_E^1 + a_E^2 x,\ a_E^3 + a_E^2 y),\ a_E^i \in \mathbb{R},\ E \in \mathcal{E}_h\},\\
W_h &= \{w \in L^2(\Omega) : w|_E \text{ is constant for all } E \in \mathcal{E}_h\},\\
L_h &= \{\mu \in L^2(\partial\mathcal{E}_h) : \mu|_e \text{ is constant},\ e \in \partial\mathcal{E}_h;\ \mu|_e = 0,\ e \subset \partial\Omega\},
\end{aligned}$$

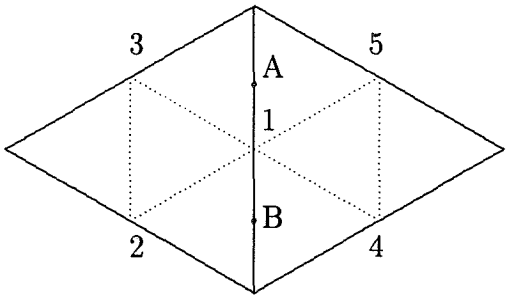


FIGURE 1. The illustration of the definition of I_k^{k-1}.

where $\partial\mathcal{E}_h$ denotes the set of all interior edges. Then the hybrid form of the mixed method for (1.1) is to seek $(\sigma_h, u_h, \lambda_h) \in \Lambda_h \times W_h \times L_h$ such that

$$\begin{aligned} \sum_{E\in\mathcal{E}_h} (\nabla\cdot\sigma_h, w)_E &= (f, w), \quad \forall\, w \in W_h, \\ (\alpha\sigma_h, v) - \sum_{E\in\mathcal{E}_h} [(u_h, \nabla\cdot v)_E - (\lambda_h, v\cdot\nu_E)_{\partial E}] &= 0, \qquad \forall\, v \in \Lambda_h, \\ \sum_{E\in\mathcal{E}_h} (\sigma_h\cdot\nu_E, \mu)_{\partial E} &= 0, \qquad \forall\, \mu \in L_h, \end{aligned} \tag{3.1}$$

where ν_E denotes the unit outer normal to E and $\alpha = a^{-1}$. The solution σ_h is introduced to approximate the vector field

$$\sigma = -a\nabla u,$$

which is the variable of primary interest in many applications. Since σ lies in the space

$$H(\mathrm{div};\Omega) = \left\{v \in (L^2(\Omega))^2 : \nabla\cdot v \in L^2(\Omega)\right\},$$

and we do not require that Λ_h be a subspace of $H(\mathrm{div};\Omega)$, the last equation in (3.1) is used to enforce that the normal components of σ_h are continuous across the interior edges in $\partial\mathcal{E}_h$, so in fact $\sigma_h \in H(\mathrm{div};\Omega)$.

There is no continuity requirement on the spaces Λ_h and W_h, so σ_h and u_h can be locally (element by element) eliminated from (3.1). In fact, from [15], (3.1) can be algebraically condensed to the symmetric, positive definite system for the Lagrange multiplier λ_h:

$$M_h\lambda_h = F_h, \tag{3.2}$$

where the contributions of the triangle E to the stiffness matrix M_h and the right-hand side F_h are

$$m_{ij}^E = \frac{\overrightarrow{\nu}_E^i\cdot\overrightarrow{\nu}_E^j}{(\alpha,1)_E}, \quad F_i^E = -\frac{(\alpha J_E^f, \overrightarrow{\nu}_E^i)_E}{(\alpha,1)_E} + (J_E^f, \nu_E^i)_{e_E^i}, \tag{3.3}$$

where ν_E^i denotes the outer unit normal to the edge e_E^i, $\overrightarrow{\nu}_E^i = |e_E^i|\nu_E^i$, $|e_E^i|$ is the length of e_E^i, $J_E^f = (f,1)_E(x,y)/(2|E|)$, and $|E|$ denotes the area of E.

Let P_h denote the $L^2(\Omega)$ projection operator onto W_h, $\alpha_h = P_h\alpha$, and $f_h = P_h f$. Also, set

$$\tilde{f}_h|_E = \frac{f_h}{2}\left(3 - \frac{\alpha}{\alpha_h}\right)|_E,$$

and

$$\tilde{a}_h(\psi,\varphi) = \sum_{E\in\mathcal{E}_h} (\alpha_h^{-1}\nabla\psi, \nabla\varphi)_E.$$

Then as shown in [15], the system (3.2) corresponds to the system arising from the triangular nonconforming finite element method: find $\psi_h \in V_h$ such that

$$\tilde{a}_h(\psi_h,\varphi) = (\tilde{f}_h,\varphi), \quad \forall\varphi \in V_h. \tag{3.4}$$

Hence Algorithm 2.1 can be used to solve (3.2), i.e., the mixed method (3.1). It should be also noted that the natural degrees of freedom, i.e., the values at the midpoint of edges, of L_h and V_h are the same.

After the computation of λ_h, σ_h and u_h (if they are needed) can be recovered as follows. Set $\sigma_h|_E = (a_E + b_E x, c_E + b_E y)$ and $\overline{f}_E = f_h|_E$. Then it follows from [15] that

$$\begin{aligned}
b_E &= \frac{\overline{f}_E}{2},\\
a_E &= -\frac{1}{(\alpha,1)_E}\left(\sum_{i=1}^3 |e_E^i|\nu_E^{i(1)}\lambda_h|_{e_E^i} + \frac{\overline{f}_E}{2}(\alpha,x)_E\right),\\
c_E &= -\frac{1}{(\alpha,1)_E}\left(\sum_{i=1}^3 |e_E^i|\nu_E^{i(2)}\lambda_h|_{e_E^i} + \frac{\overline{f}_E}{2}(\alpha,y)_E\right),
\end{aligned}$$

and

$$u_h|_E = \frac{1}{2|E|}\left((\alpha\sigma_h,(x,y))_E + \sum_{i=1}^3 \lambda_h|_{e_E^i}((x,y),\nu_E^i)_{e_E^i}\right), \quad E\in\mathcal{E}_h.$$

We now consider a modified version of the mixed method (3.1) in which the coefficient α is projected into the space W_h [20]: find $(\sigma_h, u_h, \lambda_h) \in \Lambda_h \times W_h \times L_h$ such that

$$\begin{aligned}
\sum_{E\in\mathcal{E}_h} (\nabla\cdot\sigma_h, w)_E &= (f,w), \quad \forall\, w\in W_h,\\
(\alpha_h\sigma_h, v) - \sum_{E\in\mathcal{E}_h}[(u_h, \nabla\cdot v)_E - (\lambda_h, v\cdot\nu_E)_{\partial E}] &= 0, \quad \forall\, v\in\Lambda_h,\\
\sum_{E\in\mathcal{E}_h} (\sigma_h\cdot\nu_E, \mu)_{\partial E} &= 0, \quad \forall\,\mu\in L_h.
\end{aligned} \tag{3.5}$$

Associated with this projected formulation, the linear system has the form in place of (3.3):

$$m_{ij}^E = \frac{\overline{\nu}_E^i\cdot\overline{\nu}_E^j}{(\alpha,1)_E}, \quad F_i^E = -\frac{(J_E^f,\overline{\nu}_E^i)_E}{|E|} + (J_E^f,\nu_E^i)_{e_E^i}, \quad E\in\mathcal{E}_h. \tag{3.6}$$

The corresponding nonconforming system becomes: find $\psi_h \in V_h$ such that

$$\tilde{a}_h(\psi_h,\varphi) = (f_h,\varphi), \quad \forall\varphi\in V_h, \tag{3.7}$$

The present systems in (3.6) and (3.7) are simpler than the corresponding systems in (3.3) and (3.4). The advantage of the projected mixed formulation over the usual one is more obvious for the mixed finite element method over rectangles, which will be carried out in the next section.

RECTANGULAR ELEMENTS

In this section we consider the lowest order Raviart-Thomas space over rectangles [31]. Let $\mathcal{E}_h$ be a partition of Ω into rectangles oriented along the coordinate axes, and let $Q_{i,j}(E)$ be the space of polynomials of degree not bigger than i in x and j in y. The rectangular mixed space [31] is defined by

$$\begin{aligned} V_h(E)&=Q_{1,0}(E)\times Q_{0,1}(E),\\ W_h(E)&=P_0(E),\\ L_h(e)&=P_0(e). \end{aligned}$$

We first consider the usual mixed method (3.1). For each $E\in\mathcal{E}_h$, set

$$\begin{aligned} &\alpha_E^x=\frac{(\alpha,x^2)_E}{(\alpha,x)_E}-\frac{(\alpha,x)_E}{(\alpha,1)_E}, && \alpha_E^y=\frac{(\alpha,y^2)_E}{(\alpha,y)_E}-\frac{(\alpha,y)_E}{(\alpha,1)_E},\\ &\alpha_{e_E^i}^x=\frac{(x,1)_{e_E^i}}{(\alpha,x)_E}-\frac{|e_E^i|}{(\alpha,1)_E}, && \alpha_{e_E^i}^y=\frac{(y,1)_{e_E^i}}{(\alpha,y)_E}-\frac{|e_E^i|}{(\alpha,1)_E},\\ &A_E=(\alpha,x)_E\alpha_E^x+(\alpha,y)_E\alpha_E^y. \end{aligned}$$

Then, following [15], it follows that the contributions of the rectangle E to the stiffness matrix and the right-hand side are

$$\begin{aligned} m_{ij}^E=&\frac{1}{(\alpha,1)_E}\overline{\nu}_E^i\cdot\overline{\nu}_E^j+\frac{1}{A_E}(\alpha,x)_E^2\alpha_{e_E^i}^x\alpha_{e_E^j}^x\nu_E^{i(1)}\nu_E^{j(1)}\\ &-\frac{1}{A_E}(\alpha,x)_E(\alpha,y)_E\alpha_{e_E^j}^x\alpha_{e_E^i}^y\nu_E^{i(2)}\nu_E^{j(1)}\\ &-\frac{1}{A_E}(\alpha,x)_E(\alpha,y)_E\alpha_{e_E^i}^x\alpha_{e_E^j}^y\nu_E^{i(1)}\nu_E^{j(2)}\\ &+\frac{1}{A_E}(\alpha,y)_E^2\alpha_{e_E^i}^y\alpha_{e_E^j}^y\nu_E^{i(2)}\nu_E^{j(2)},\\ F_i^E=&\frac{\overline{f}_E}{A_E}(\alpha,x)_E(\alpha,y)_E\left(\alpha_E^y\alpha_{e_E^i}^x\nu_E^{i(1)}+\alpha_E^x\alpha_{e_E^i}^y\nu_E^{i(2)}\right). \end{aligned}\tag{4.1}$$

Namely, we again have the linear system (3.2) for the Lagrange multiplier λ_h for the rectangular elements. Also, after the calculation of λ_h, we can compute σ_h and u_h as

follows. For each $E \in \mathcal{E}_h$, let $\sigma_h|_E = (a_E + b_E x, c_E + d_E y)$. Then we have

$$\begin{aligned}
a_E &= \left\{\sum_{i=1}^4 \left\{ \left((\alpha,x)_E^2 \alpha^x_{e_E^i} - A_E|e_E^i|\right) \nu_E^{i(1)} - (\alpha,x)_E(\alpha,y)_E \alpha^y_{e_E^i} \nu_E^{i(2)} \right\} \lambda_h|_{e_E^i} \right. \\
&\quad \left. -(\alpha,x)_E(\alpha,y)_E \alpha_E^y \overline{f}_E \right\} / \left((\alpha,1)_E A_E\right), \\
c_E &= \left\{\sum_{i=1}^4 \left\{ \left((\alpha,y)_E^2 \alpha^y_{e_E^i} - A_E|e_E^i|\right) \nu_E^{i(2)} - (\alpha,x)_E(\alpha,y)_E \alpha^x_{e_E^i} \nu_E^{i(1)} \right\} \lambda_h|_{e_E^i} \right. \\
&\quad \left. -(\alpha,x)_E(\alpha,y)_E \alpha_E^x \overline{f}_E \right\} / \left((\alpha,1)_E A_E\right), \\
b_E &= \tfrac{1}{A_E} \sum_{i=1}^4 \left(-(\alpha,x)_E \alpha^x_{e_E^i} \nu_E^{i(1)} + (\alpha,y)_E \alpha^y_{e_E^i} \nu_E^{i(2)} \right) \lambda_h|_{e_E^i} + \tfrac{1}{A_E}(\alpha,y)_E \alpha_E^y \overline{f}_E, \\
d_E &= \tfrac{1}{A_E} \sum_{i=1}^4 \left((\alpha,x)_E \alpha^x_{e_E^i} \nu_E^{i(1)} - (\alpha,y)_E \alpha^y_{e_E^i} \nu_E^{i(2)} \right) \lambda_h|_{e_E^i} + \tfrac{1}{A_E}(\alpha,x)_E \alpha_E^x \overline{f}_E,
\end{aligned}$$

and

$$u_E = \frac{(\alpha,x)_E(\alpha,y)_E}{|E|A_E} \left\{ \sum_{i=1}^4 \left(\alpha_E^y \alpha^x_{e_E^i} \nu_E^{i(1)} + \alpha_E^x \alpha^y_{e_E^i} \nu_E^{i(2)} \right) \lambda_h|_{e_E^i} + \alpha_E^x \alpha_E^y \overline{f}_E \right\}.$$

We now consider the projected mixed method (3.5) on rectangles. For each $E \in \mathcal{E}_h$, let $|\nu_E^i|' = |\nu_E^{i(1)}| - |\nu_E^{i(2)}|$, and let Δx_E and Δy_E denote the x-length and the y-length of E, respectively. Then (4.1) reduces to

$$\begin{aligned}
m_{ij}^E &= \tfrac{1}{(\alpha,1)_E} \overline{\nu}_E^i \cdot \overline{\nu}_E^j + \tfrac{3|E|^2}{R_E(\alpha,1)_E} |\nu_E^i|' |\nu_E^j|', \\
F_i^E &= -\tfrac{(J_E^f, \overline{\nu}_E^i)_E}{|E|} + (J_E^f, \nu_E^i)_{e_E^i},
\end{aligned}$$

where

$$R_E = \Delta x_E^2 + \Delta y_E^2, \quad J_E^f = \frac{\overline{f}_E}{R_E} \left(\Delta y_E^2 x, \Delta x_E^2 y \right).$$

The σ_h and u_h can be computed in a much simpler way. Let $(\bar{x}_E, \bar{y}_E)$ denote the center of the rectangle E. Then we have

$$\begin{aligned}
a_E &= \tfrac{|E|}{(\alpha,1)_E} \sum_{i=1}^4 \left\{ \tfrac{6\bar{x}_E}{R_E} \left(|\nu_E^{i(1)}| - |\nu_E^{i(2)}| \right) - \tfrac{1}{\Delta x_E} \nu_E^{i(1)} \right\} \lambda_h|_{e_E^i} - \tfrac{\bar{x}_E \Delta y_E^2 \overline{f}_E}{R_E}, \\
b_E &= \tfrac{6|E|}{(\alpha,1)_E R_E} \sum_{i=1}^4 \left(-|\nu_E^{i(1)}| + |\nu_E^{i(2)}| \right) + \tfrac{\Delta y_E^2 \overline{f}_E}{R_E}, \\
c_E &= \tfrac{|E|}{(\alpha,1)_E} \sum_{i=1}^4 \left\{ \tfrac{6\bar{y}_E}{R_E} \left(-|\nu_E^{i(1)}| + |\nu_E^{i(2)}| \right) - \tfrac{1}{\Delta y_E} \nu_E^{i(2)} \right\} \lambda_h|_{e_E^i} - \tfrac{\bar{y}_E \Delta x_E^2 \overline{f}_E}{R_E}, \\
d_E &= \tfrac{6|E|}{(\alpha,1)_E R_E} \sum_{i=1}^4 \left(|\nu_E^{i(1)}| - |\nu_E^{i(2)}| \right) + \tfrac{\Delta x_E^2 \overline{f}_E}{R_E},
\end{aligned}$$

and

$$u_h|_E = \frac{1}{2R_E} \sum_{i=1}^4 \left(\Delta y_E^2 |\nu_E^{i(1)}| + \Delta x_E^2 |\nu_E^{i(2)}| \right) \lambda_h|_{e_E^i} + \frac{(\alpha,1)_E |E|}{12R_E} \overline{f}_E.$$

Now we see that the projected mixed method produces a much simpler system. The equivalence between the mixed method and the nonconforming method can be established as in the triangular case [15]. In the present case, the corresponding nonconforming space is

$$N_h = \Big\{ \xi : \xi|_E = a_E^1 + a_E^2 x + a_E^3 y + a_E^4 (x^2 - y^2),\ a_E^i \in \mathbb{R},\ \forall E \in \mathcal{E}_h;$$
$$\text{if } E_1 \text{ and } E_2 \text{ share an edge } e, \text{ then } \int_e \xi|_{\partial E_1}\, ds = \int_e \xi|_{\partial E_2}\, ds;$$
$$\text{and } \int_{\partial E \cap \partial\Omega} \xi|_{\partial\Omega}\, ds = 0 \Big\}.$$

Moreover, the definition of Algorithm 2.1 remains the same here provided that a coarse-to-fine intergrid transfer operator can be defined for the rectangular elements. As an example, we give a variant of the operator in Example 1. Other cases can be similarly extended.

Let $\{\mathcal{E}_{h_k}\}_{k=0}$ be a family of triangulations of Ω such that $\mathcal{E}_{h_k} = \mathcal{E}_k$ is constructed by connecting the midpoints of the edges of the rectangles in $\mathcal{E}_{k-1}$. Following [1], we define the coarse-to-fine intergrid transfer operators $I_{k-1}^k : V_{k-1} \to V_k$ as follows. If $\xi \in V_{k-1}$ and e is an edge of a rectangle in $\mathcal{E}_k$, then $I_{k-1}^k \xi \in V_k$ is defined by

$$\frac{1}{|e|}\int_e I_k v ds = \begin{cases} 0 & \text{if } e \subset \partial\Omega, \\ \dfrac{1}{|e|}\displaystyle\int_e v ds & \text{if } e \not\subset \partial E \text{ for any } E \in \mathcal{E}_{k-1}, \\ \dfrac{1}{2|e|}\displaystyle\int_e (v|_{E_1} + v|_{E_2}) ds & \text{if } e \subset \partial E_1 \cap \partial E_2 \text{ for some } E_1, E_2 \in \mathcal{E}_{k-1}. \end{cases}$$

The results in the previous theorem remain the same here [18]. We remark that the analysis in the paper applies to differential problems with a tensor coefficient and a lower order term. Also, while we only considered the Raviart-Thomas spaces on triangles and rectangles, other mixed finite element spaces (see, e.g., [11, 12, 13, 19, 22, 28, 29, 31]) can be similarly dealt with. For more information on these extensions, refer to [15, 17, 16, 18]. Finally, refer to [10, 23, 24, 25, 26, 30, 32, 34, 35] for multigrid algorithms for mixed finite element methods using different approaches than the present one.

NUMERICAL EXPERIMENTS

We present the results of a couple of numerical examples to illustrate the theory developed in the earlier sections and to show a comparison between the results obtained here and those generated by the well established conforming finite element and finite difference multigrid algorithms [7, 8]. Thus we apply the numerical data given in these earlier papers. These results are reported in [18]; more numerical results can be found in [15]. Numerical experiments for comparisons among the operators described in section two will be for future work.

EXAMPLE 1. In the first example we consider the Laplace equation on the unit square

$$\begin{aligned} -\Delta u &= f \quad \text{in } \Omega = (0,1)^2, \\ u &= 0 \quad \text{on } \partial\Omega. \end{aligned} \tag{5.1}$$

h_K	(κ_v, δ_v)	(κ_w, δ_w)	$(\kappa_{vv}, \delta_{vv})$
1/8	(1.48, .43)	(1.46, .40)	(1.45, .39)
1/16	(1.64, .46)	(1.47, .42)	(1.47, .41)
1/32	(1.81, .50)	(1.48, .43)	(1.48, .43)
1/64	(1.86, .51)	(1.48, .43)	(1.48, .43)
1/128	(1.96, .54)	(1.48, .43)	(1.48, .43)

Table 1. Convergence Results for Example 1

We approximate the solution to (5.1) using the triangular nonconforming method (i.e., the triangular mixed method). The analysis of section two guarantees that the condition number of $B_K A_K$ for the variable $\mathcal{V}$-cycle algorithm can be bounded independently on the number of levels and the $\mathcal{W}$-cycle algorithm has an optimal convergence property. Table 1 gives the condition number κ for the system $B_K A_K$ and the reduction factor for the system $I - B_K A_K$ as a function of the mesh size on the finest grid, where the $\mathcal{V}$-cycle, $\mathcal{W}$-cycle, and variable $\mathcal{V}$-cycle algorithms are indicated by (κ_v, δ_v), (κ_w, δ_w), and $(\kappa_{vv}, \delta_{vv})$, respectively. The $\mathcal{V}$-cycle and $\mathcal{W}$-cycle schemes use one smoothing step. The coarse-to-fine intergrid transfer operator in Example 1 of section two is used. (To see how the convergence rate depends upon the number of the smoothing steps, refer to [15].) For all of the runs, the coarse grid is of size $h_0 = 1/2$. As noticed in the conforming case [8], the variable $\mathcal{V}$-cycle and the $\mathcal{W}$-cycle algorithms have essentially identical computational results. This is due to the fact that both algorithms have exactly the same number of total smoothings on each grid in the multi-level iteration. While there is no complete theory for the $\mathcal{V}$-cycle algorithm with the averaging transfer operator, it is of practical interest that the condition numbers for this cycle remain relatively small, but the convergence rate deteriorates with the mesh size. Finally, compared with the numerical results obtained in [7, 8], we see that the nonconforming multigrid algorithms in fact compare favorably with these standard multigrid algorithms.

EXAMPLE 2. In the second example we consider the following model problem with a variable coefficient $a(x)$:

$$\begin{aligned} -\nabla \cdot (a \nabla u) &= f \quad \text{in } \Omega = (0,1)^2, \\ u &= 0 \quad \text{on } \partial\Omega. \end{aligned}$$

Similar results as in Table 1 are obtained for this problem using the triangular elements. Hence we examine the rectangular elements. This example uses the same set of data as the first example does. The numerical results are shown in Table 2. The same facts as in the first example are also observed here.

h_K	(κ_v, δ_v)	(κ_w, δ_w)	$(\kappa_{vv}, \delta_{vv})$
1/8	(1.54, .44)	(1.50, .42)	(1.51, .42)
1/16	(1.65, .46)	(1.52, .44)	(1.52, .43)
1/32	(1.86, .51)	(1.53, .45)	(1.53, .45)
1/64	(1.95, .53)	(1.53 .45)	(1.53, .45)
1/128	(2.07, .60)	(1.53, .45)	(1.53, .45)

Table 2. Convergence Results for Example 2

REFERENCES

[1] T. Arbogast and Zhangxin Chen, On the implementation of mixed methods as nonconforming methods for second order elliptic problems, *Math. Comp.*, **64** (1995), to appear.

[2] D. Arnold and F. Brezzi, Mixed and nonconforming finite element methods: implementation, postprocessing and error estimates, *RAIRO Model. Math. Anal. Numer.*, **19** (1985), 7–32.

[3] R. Bank and C. Douglas, Sharp estimates for multigrid rates of convergence with general smoothing and acceleration, *SIAM J. Numer. Anal.*, **22** (1985), 617–633.

[4] R. Bank and T. Dupont, An optimal order process for solving finite element equations, *Math. Comp.*, **36** (1981), 35–51.

[5] D. Braess and W. Hackbusch, A new convergence proof for the multigrid including the $\mathcal{V}$-cycle, *SIAM J. Numer. Anal.*, **20** (1983), 967–975.

[6] D. Braess and R. Verfürth, Multigrid methods for nonconforming finite element methods, *SIAM J. Numer. Anal.*, **27** (1990), 979–986.

[7] J. Bramble, R. Ewing, J. Pasciak, and J. Shen, The analysis of multigrid algorithms for cell centered finite difference methods, Preprint, to appear in *Advances in Computational Math.*, 1995.

[8] J. Bramble, J. Pasciak, and J. Xu, The analysis of multigrid algorithms with non-nested spaces or non-inherited quadratic forms, *Math. Comp.*, **56** (1991), 1–34.

[9] S. Brenner, An optimal-order multigrid method for P1 nonconforming finite elements, *Math. Comp.*, **52** (1989), 1–15.

[10] S. Brenner, A multigrid algorithm for the lowest-order Raviart-Thomas mixed triangular finite element method, *SIAM Numer. Anal.*, **29** (1992), 647–678.

[11] F. Brezzi, J. Douglas, Jr., R. Durán, and M. Fortin, Mixed finite elements for second order elliptic problems in three variables, *Numer. Math.*, **51** (1987), 237–250.

[12] F. Brezzi, J. Douglas, Jr., M. Fortin, and L. D. Marini, Efficient rectangular mixed finite elements in two and three space variables, *RAIRO Modél. Math. Anal. Numér.*, **21** (1987), 581–604.

[13] F. Brezzi, J. Douglas, Jr., and L. D. Marini, Two families of mixed finite elements for second order elliptic problems, *Numer. Math.*, **47** (1985), 217–235.

[14] Zhangxin Chen, Analysis of mixed methods using conforming and nonconforming finite element methods, *RAIRO Math. Model. Numer. Anal.*, **27** (1993), 9–34.

[15] Zhangxin Chen, Equivalence between and multigrid algorithms for mixed and nonconforming methods for second order elliptic problems, IMA Preprint #1218, 1994.

[16] Zhangxin Chen, R. Ewing, Y. Kuznetsov, R. Lazarov, and S. Maliassov, Multilevel preconditioners for mixed methods for second order elliptic problems, *ISC-94-10-MATH Technical Report, Texas A&M University.*

[17] Zhangxin Chen, R. Ewing, and R. Lazarov, Domain decomposition algorithms for mixed methods for second order elliptic problems, *Math. Comp.*, **65** (1996), to appear.

[18] Zhangxin Chen, D. Kwak, and Y. Yon, Multigrid algorithms for nonconforming and mixed methods for symmetric and nonsymmetric problems, IMA Preprint #1277, 1994.

[19] Zhangxin Chen and J. Douglas, Jr., Prismatic mixed finite elements for second order elliptic problems, *Calcolo*, **26** (1989), 135–148.

[20] Zhangxin Chen and J. Douglas, Jr., Approximation of coefficients in hybrid and mixed methods for nonlinear parabolic problems, *Mat. Aplic. Comp.*, **10** (1991), 137–160.

[21] L. Cowsar, Domain decomposition methods for nonconforming finite element spaces of Lagrange type, *Proceedings of the Sixth Copper Mountain Conference on Multigrid Methods*, N. Melson et al., eds., NASA Conference Publication 3224, Part 1, 1993, 93–109.

[22] J. Douglas and J. Wang, A new family of mixed finite element spaces over rectangles, *Comp. Appl. Math.*, **12** (1993), 183–197.

[23] R.E. Ewing, Yu. Kuznetsov, R. Lazarov, and S. Maliassov, Preconditioning of nonconforming finite element approximations of second order elliptic problems, In *Proceedings of the Third International Conf. on Advances in Numer. Meth. and Appl.*, I. Dimov, Bl. Sendov, and P. Vassilevski, eds., World Scientific, London & Hong Kong (1994), 101–110.

[24] R.E. Ewing, Yu. Kuznetsov, R. Lazarov, and S. Maliassov, Substructuring preconditioning for finite element approximations of second order elliptic problems. I. Nonconforming linear elements for the Poisson equation in parallelepiped, ISC Preprint #2, TAMU, 1994.

[25] R. Ewing and M. Wheeler, Computational aspects of mixed finite element methods, In: *Numerical Methods for Scientific Computing*, R.S. Stepleman, ed., North-Holland, New York, 1983, 163–172.

[26] C. Lee, A nonconforming multigrid method using conforming subspaces, *Proceedings of the Sixth Copper Mountain Conference on Multigrid Methods*, N. Melson et al., eds., NASA Conference Publication 3224, Part 1, 1993, 317–330.

[27] L. D. Marini, An inexpensive method for the evaluation of the solution of the lowest order Raviart–Thomas mixed method, *SIAM J. Numer. Anal.*, **22** (1985), 493–496.

[28] J. C. Nedelec, Mixed finite elements in $\mathbf{R}^3$, *Numer. Math.*, **35** (1980), 315–341.

[29] J. C. Nedelec, A new family of mixed finite elements in $\mathbf{R}^3$, *Numer. Math.*, **50** (1986), 57–81.

[30] P. Peisker, A multigrid algorithm for the biharmonic problem, *Numer. Math.*, **46** (1985), 623–634.

[31] P.A. Raviart and J.M. Thomas, A mixed finite element method for second order elliptic problems, In: *Mathematical aspects of the FEM, Lecture Notes in Mathematics*, **606**, Springer-Verlag, Berlin & New York (1977), 292–315.

[32] V. Shajdurov, A multigrid iterative algorithm for the mixed finite element method, *Soviet J. Numer. Anal. Math. Modelling*, **3** (1988), 231–243.

[33] P. Vassilevski and J. Wang, An application of the abstract multilevel theory to nonconforming finite element methods, *SIAM J. Numer. Anal.*, **32** (1994), 235–248.

[34] R. Verfürth, A multilevel algorithm for mixed problems, *SIAM J. Numer. Anal.*, **21** (1984), 264–271.

[35] R. Verfürth, Multilevel algorithms for mixed problems II. Treatment of the mini-element, *SIAM J. Numer. Anal.*, **25** (1988), 285–293.

Page intentionally left blank

EFFECTIVE NUMERICAL METHODS FOR SOLVING ELLIPTIC PROBLEMS IN STRENGTHENED SOBOLEV SPACES

Eugene G. D'yakonov
Department of Computer Mathematics and Cybernetics
Moscow State University
Moscow, 119899, Russia

SUMMARY

Fourth-order elliptic boundary value problems in the plane can be reduced to operator equations in Hilbert spaces G that are certain subspaces of the Sobolev space $W_2^2(\Omega) \equiv G^{(2)}$. Appearance of asymptotically optimal algorithms for Stokes type problems made it natural to focus on an approach that considers $\mathrm{rot}\, w \equiv [D_2 w, -D_1 w] \equiv \vec{u}$ as a new unknown vector–function, which automatically satisfies the condition $\mathrm{div}\, \vec{u} = 0$. In this work, we show that this approach can also be developed for an important class of problems from the theory of plates and shells with stiffeners. The main mathematical problem was to show that the well-known inf-sup condition (normal solvability of the divergence operator) holds for special Hilbert spaces. This result is also essential for certain hydrodynamics problems.

1. INTRODUCTION

Fourth-order elliptic boundary value problems can be reduced to operator equations in Hilbert spaces G that are certain subspaces of the Sobolev space $W_2^2(\Omega) \equiv G^{(2)}$. Construction of asymptotically optimal grid approximations and, most particularly, asymptotically optimal algorithms are very difficult now because the associated spline subspaces are not of Lagrangian type. These difficulties evoked a series of attempts to reduce such problems to second-order differential equations, but with no essential progress in the construction of asymptotically optimal algorithms.

Appearance of asymptotically optimal algorithms for Stokes type problems made it natural to focus on an approach that considers

$$[u_1, u_2] \equiv \vec{u} \equiv [\frac{\partial w}{\partial x_2}, -\frac{\partial w}{\partial x_1}] \equiv [D_2 w, -D_1 w] \equiv \text{rot } w \tag{1.1}$$

as a new unknown vector–function, which automatically satisfies the condition div $\vec{u} = 0$ (see [1–9]). (This condition explains why we prefer to use rot w instead of grad w.)

In what follows, we assume, for simplicity, that Ω is a simply connected domain with Lipschitz piecewise smooth boundary Γ. We suppose that

$$W \equiv W_2^2(\Omega; \Gamma^0)$$

consists of $w \in G^{(2)}$ that, with their first derivatives, vanish on the set $\Gamma^0 \subset \Gamma$, where one-dimensional measures of Γ^0 and $\Gamma^1 \equiv \Gamma \setminus \Gamma^0$ are positive and Γ_0 is a connected arc.

We start by considering classical variational problems (plates without stiffeners), that deal with variational problem of finding

$$w = \text{argmin } \Phi(w), \tag{1.2}$$

where the energy functional is defined by

$$\Phi(w) \equiv I_2(w) - 2l(w), \tag{1.3}$$

$$I_2(w) \equiv \sum_{s=1}^{2} \sum_{r=1}^{2} (a_{s,r}, (D_s D_r w)^2)_0 + 2(a_0, D_1^2 w D_2^2 w)_0, \tag{1.4}$$

the conditions

$$\left.\begin{array}{c} a_{1,2} = a_{2,1}, \quad a_{s,r}(x) \geq \kappa_0 > 0, \ s = 1, 2, \ r = 1, 2, \\ a_{1,1}(x) a_{2,2}(x) - a_0^2(x) \geq \kappa_1 > 0, \ \forall x \in \Omega, \end{array}\right\} \tag{1.5}$$

are satisfied, and

$$l(w) = (f_{1,1}, D_2 w)_0 - (f_{1,2}, D_1 w)_0. \tag{1.6}$$

Here, $(u, v)_0 \equiv (u, v)_{L_2(\Omega)}$ and $f_{1,r} \in L_2(\Omega), r = 1, 2$.

Next, we consider a subset S of $\bar{\Omega}$ consisting of straight line segments (stiffeners or stringers) $S_1, \ldots, S_m$. For simplicity, we assume that the end points of each stiffener belong to Γ. Thus (considered as cuttings lines), they define a partition of $\bar{\Omega}$ into a set of blocks (panels) $P_1, \ldots, P_{m'}$. We also assume that, if an inner point of a S_r belongs to Γ, then S_r belongs to Γ^1 (note that $m' = 1$ if $S \subset \Gamma$). ($\Gamma' \equiv \Gamma \cup S$ corresponds to the union of the panel boundaries.) We replace $I_2(w)$ by

$$\bar{I}_2(w) \equiv I_2(w) + \sum_{r=1}^{m} \int_{S_r} [c_{r,1}(D_s^2 w)^2 + c_{r,2}(D_s D_n w)^2] ds, \tag{1.7}$$

where $c_{r,1}$ and $c_{r,2}$ are positive constants ($r \in [1,m]$), s and $n \equiv \vec{n}$ refer to the respective arclength parameter and normal with respect to $S_r, r \in [1,m]$, and the Hilbert space W consists of functions in $W_2^2(\Omega;\Gamma^0)$ with special traces of $D_s w$ and $D_n w$ on each S_r. These traces must belong to $W_2^1(S_r), r \in [1,m]$, so we may define the inner product $(w,w')_W$ by

$$(w,w')_{2,\Omega} + \sum_{r=1}^{m}[(c_{r,1}, D_s^2 w D_s^2 w')_{0,S_r} + (c_{r,2}, D_s D_n w D_s D_n w')_{0,S_r}]. \qquad (1.8)$$

If the end points of a stiffener S_r belong to Γ^0, then these traces must belong to $\overset{o}{W}{}_2^2(S_r)$. The case with only one end point of S_r on Γ^0 is fairly similar. Also, we may replace $l(w)$ in (2.8) by

$$\bar{l}(w) \equiv l(w) + \sum_{r=1}^{m}[(f'_{r,1}, D_s^2 w)_{0,S_r} + (f'_{r,2}, D_s D_n w)_{0,S_r}], \qquad (1.9)$$

where $f'_{r,1} \in L_2(S_r), f'_{r,2} \in L_2(S_r),\ r \in [1,m]$. This implies that we deal with the original variational problem

$$w = \arg \min_{w' \in W}[\bar{I}_2(w') - 2\bar{l}(w')]. \qquad (1.10)$$

First use of analogous problems in pre-Hilbert spaces dates back to the paper of S. Timoshenko in 1915; see also [10,11].

2. REDUCTION TO STOKES TYPE SYSTEMS

Let $\vec{s} \equiv \vec{s}_r \equiv [\cos\alpha_r, \sin\alpha_r]$ determine the direction of $S_r, r \in [1,m]$. Then

$$\vec{n} \equiv \vec{n}_r \equiv [-\sin\alpha_r, \cos\alpha_r]$$

and, in accordance with (1.1), on S_r, we have

$$D_s w = -\cos\alpha_r u_2 + \sin\alpha_r u_1 \equiv I_{r,s}(\vec{u})$$

and

$$D_n w = \sin\alpha_r u_2 + \cos\alpha_r u_1 \equiv I_{r,n}(\vec{u}),\ r \in [1,m].$$

With the Hilbert space W in (1.10), we associate a Hilbert space rot W. This we describe by introducing a Hilbert space $G_1 \subset (W_2^1(\Omega;\Gamma^0))^2$, whose elements

are vector fields $\vec{u}$ belonging to $(W_2^1(\Omega;\Gamma^0))^2$ and such that the traces of $I_{r,s}(\vec{u})$ and $I_{r,s}(\vec{u})$ on S_r (they exist in the sense of traces of functions in $W_2^1(\Omega)$) satisfy

$$I_{r,s}(\vec{u}) \in W_2^1(S_r),\ I_{r,n}(\vec{u}) \in W_2^1(S_r),\ r \in [1,m]. \tag{2.1}$$

The inner product in G_1 is defined by

$$(\vec{u},\vec{v})_{G_1} \equiv (\vec{u},\vec{v})_{1,\Omega}$$
$$+\sum_{r=1}^{m}[(1, I_{r,s}(\vec{u})I_{r,s}(\vec{v}))_{1,S_r} + (1, I_{r,n}(\vec{u})I_{r,n}(\vec{v}))_{1,S_r}] \tag{2.2}$$

(if the end points of a stiffener S_r belong to Γ^0, then the above traces must belong to $(\overset{o}{W}{}_2^1\ (S_r))^2$; the case with only one end point of S_r on Γ^0 is fairly similar). Then rot $W \subset G_1$ is a subspace of solenoidal vector fields.

We replace (1.10) by the problem of finding $u \in G \equiv G_1 \times G_2$ (with $G_2 \equiv L_2(\Omega)$) such that

$$\left.\begin{aligned} \bar{b}_{1,1}(u_1;u_1') + b_{1,2}(u_2;u_1') = \bar{l}_1(u_1'),\ \forall u_1' \in G_1 \\ b_{2,1}(u_1;u_2') = 0,\ \forall u_2' \in G_2, \end{aligned}\right\} \tag{2.3}$$

where

$$\bar{b}_{1,1}(u_1;u_1') \equiv b_{1,1}(u_1;u_1')+$$
$$\sum_{r=1}^{m}[c_{r,1}(1, I_{r,s}(\vec{u})I_{r,s}(\vec{u}'))_{1,S_r} + c_{r,2}(1, I_{r,n}(\vec{u})I_{r,n}(\vec{u}'))_{1,S_r}] \tag{2.4}$$

and

$$\bar{l}_1(u_1') \equiv (f_{1,1}, u_{1,1}')_0 + (f_{1,2}, u_{1,2}')_0$$
$$+\sum_{r=1}^{m}[(f_{r,1}', D_s I_{r,s}(\vec{u}'))_{0,S_r} + (f_{r,2}', D_s I_{r,n}(\vec{u}'))_{0,S_r}].$$

Here $b_{1,1}(u_1;u_1')$ is the bilinear form associated with the case where $S = \emptyset$ and

$$b_{2,1}(u_1;u_2') \equiv (\text{div } u_1, u_2')_0 = b_{1,2}(u_2';u_1).$$

The following lemma is fundamental (necessary proofs can be found in [9]).

Lemma 2.1. *Let P be a domain with piecewise smooth boundary ∂P. Suppose that ∂P contains a straight line segment $\Gamma^*(P) \equiv S^*$ and let $\Gamma^0(P) \equiv \partial P \setminus S^*$. Suppose also that the Hilbert space $G_1(P)$ is defined as in (2.2) with only one stiffener S^*. Then, a constant K^* and $\vec{v}^* \in G_1(P)$ exist such that*

$$(\vec{v}^*, \vec{n})_{0,S^*} = 1$$

and

$$[|\vec{v}^*|_{1,P}^2 + |D_s\vec{v}^*|_{0,S^*}^2]^{1/2} \le K^*|\text{div } \vec{v}^*|_{0,P}. \tag{2.5}$$

Theorem 2.1. *Let the Hilbert space G_1 be defined as above and let $G_2 = L_2(\Omega)$. Suppose that $S \subset \Gamma^1$. Then there exists $\sigma_0 > 0$ such that*

$$\sup_{\vec{u} \in G_1} \frac{(div\ \vec{u}, p)_{0,\Omega}}{\|\vec{u}\|_{G_1}} \geq \sigma |p|_{0,\Omega},\ \sigma > 0,\ \forall p \in G_2 \tag{2.6}$$

holds.

The obtained result is a generalization of the well-known inf-sup condition (see cite[6,8–10,12,13]; it is interesting that first attempts to analyze relevant problems were made in [14]). We note also that (2.6) can be written in the form

$$\|\mathrm{div}^\dagger\| \leq \sigma^{-1}.$$

Theorem 2.2. *Let the Hilbert space G_1 be defined as above and $G_2 = L_2(\Omega)$. Suppose also that the partition of $\bar{\Omega}$ into a set of panels $P_1, \ldots, P_m$ is such that each pair P_i and P_{i+1}, $i \in [1, m-1]$, has a common side $S^*_{i,i+1} \in S$ and P_m has a side on Γ^1 (which might belong to S). Then there exists $\sigma_0 > 0$ such that* (2.6) *holds.*

Theorem 2.3. *Consider variational problem* (1.10) *replaced by* (2.3). *Suppose that S is such that the respective spaces G_1 and G_2 lead to* (2.6). *Then the rotor of the solution of* (1.10) *is the first component of the solution of* (2.3).

Similar results hold for the more difficult problem that differs from (2.3) in choices of G_1 and G_2. Elements of $G_1 \equiv G_1^0 \equiv (\overset{\circ}{W}{}_2^1(\Omega))^2$ are vector fields $\vec{u}$, belonging to $(\overset{\circ}{W}{}_2^1(\Omega))^2$, such that the traces of $I_{r,s}(\vec{v})$ and $I_{r,s}(\vec{u})$ on S_r satisfy (2.1); the inner product in G_1 is defined by (2.2); and $G_2 \equiv L_2(\Omega) \setminus 1 \equiv G_2^0$. This problem is associated with (1.10) under the choice $W = (\overset{\circ}{W}{}_2^2(\Omega))^2$ (the inner product is defined by (2.2)).

3. PROJECTIVE-GRID (MIXED FINITE ELEMENT) METHODS

We confine ourselves to domains such that Γ is a closed broken line. We can then apply triangulations $T_h(\bar{\Omega})$ (possibly composite with a finite number of the levels of local refinement) and make use of spline spaces $\hat{G}_1 \equiv \hat{G}_{1,h}$ and $\hat{G}_2 \equiv \hat{G}_{2,h}$. Here $\hat{G}_2$ consists of piecewise constant functions with respect to the triangles $T \in T_h(\bar{\Omega})$ (or augmented triangles in case of composite triangulations with local refinement like indicated in [8]); $\hat{G}_1$ consists of piecewise linear functions with respect to the triangles $T_{1/2} \in T_{h/2}(\bar{\Omega})$, where this triangulation

is a refinement of $T_h(\bar{\Omega})$ with ratio 2. Note that there is nothing essentially new in construction of subspaces $\hat{G}_1$ and $\hat{G}_2$ in comparison with the case where $S = \emptyset$ because all elements of the old $\hat{G}_1$ belong to the new one. (Of course, it is natural to construct original triangulations by refinement of the original panels, so we assume that triangulations $T_h(\bar{\Omega})$ yield triangulations $T_h(\bar{P}_i)$ for all panels $P_1, \ldots, P_{m'}$. We also assume that Γ_0 is a union of some sides of triangles $T_h \in T_h(\bar{\Omega})$).

Theorem 3.1. *For the spline spaces $\hat{G}_1$ and $\hat{G}_2$, there exists a constant σ_0^*, independent of h such that*

$$\sup_{\vec{u} \in \hat{G}_1} \frac{(\operatorname{div} \vec{u}, \hat{p})_{0,\hat{\Omega}_h}}{\|\vec{u}\|_{\hat{G}_1}} \geq \sigma_0^* |\hat{p}|_{0,\hat{\Omega}_h}, \ \sigma_0^* > 0, \ \forall \hat{p} \in \hat{G}_2. \tag{3.1}$$

Now it is clear that the convergence of our PGMs can be analyzed in accordance with the well-known theory. It is natural to make assumptions of the form

$$\|u_1\|_{1+\gamma,P_i} \leq K_{1,i}^*, \tag{3.2}$$

$$\|I_{r,n}(u_1)\|_{1+\gamma,S_r} \leq K_{r,n}, \quad \|I_{r,s}(u_1)\|_{1+\gamma,S_r} \leq K_{r,s}, \tag{3.3}$$

and

$$\|u_2\|_{\gamma,P_i} \leq K_{2,i}, \tag{3.4}$$

where $i = 1, \ldots, m', r = 1, \ldots, m$, and $\gamma \in (0, 1]$. Then it is easy to prove that asymptotic approximation properties of the strengthened Sobolev spaces are the same, and we can obtain the error estimates

$$\|\hat{u}_1 - u_1\|_{G_1} + \|\hat{u}_2 - u_2\|_0 \leq Kh^\gamma. \tag{3.5}$$

4. MULTIGRID CONSTRUCTION OF ASYMPTOTICALLY OPTIMAL PRECONDITIONERS

Our PGM yields grid systems of type

$$Lu \equiv \begin{bmatrix} L_{1,1} & L_{1,2} \\ L_{2,1} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u_1} \\ \mathbf{u_2} \end{bmatrix} = \begin{bmatrix} \mathbf{f_1} \\ 0 \end{bmatrix}, \tag{4.1}$$

such that $(L_{1,1}\mathbf{u}_1, \mathbf{v}_1)_{H_1} = \bar{b}_{1,1}(\hat{u}_1, \hat{v}_1)$, for all $\mathbf{u}_1 \in H_1$ and $\mathbf{v}_1 \in H_1$.

Here H_1 and H_2 are standard Euclidean spaces associated with $\hat{G}_1$ and $\hat{G}_2$, respectively ($\dim H_1 = \dim \hat{G}_1$, $\dim H_2 = \dim \hat{G}_2$); the operator $L_{2,1}$ is such that

$$(L_{2,1}\mathbf{u_1}, \mathbf{v_2}) = (\text{div } \hat{u}_1, \hat{v}_2)_{0,\Omega_h}, \quad \forall \hat{u}_1 \in \hat{G}_1, \forall \hat{u}_2 \in \hat{G}_2,$$

and $L_{1,2} = L_{2,1}^*$.

The resulting system differs from the case where $S = \emptyset$ only for the nodes on S. We confine ourselves to the case where

$$\bar{b}_{1,1}(u_1, v_1) = (\vec{u}, \vec{v})_{G_1}$$

and $(\vec{u}, \vec{v})_{G_1}$ is just

$$(\vec{u}, \vec{v})_{1,\Omega} + \sum_{r=1}^{m} [c_{r,1}(1, I_{r,s}(\vec{u}) I_{r,s}(\vec{v}))_{1,S_r} + c_{r,2}(1, I_{r,n}(\vec{u}) I_{r,n}(\vec{v}))_{1,S_r}]. \tag{4.2}$$

Here, $c_{r,1}$ and $c_{r,2}$ are nonnegative numbers, $r \in [1, m]$, and

$$(u, v)_{1,\Omega} \equiv \sum_{k=1}^{2} (D_k u, D_k v)_{0,\Omega}.$$

We assume that we deal with standard nested triangulations

$$T^{(l+1)}(\bar{\Omega}) \equiv T^{(l+1)}$$

of levels $l + 1 = 1, \ldots, p$, where $T^{(0)}$ is the coarse triangulation, $T^{(p)} = T_{h/2}(\bar{\Omega})$, and refinement ratio is 2.

With each triangulation $T^{(l)}$ we associate a standard finite element subspace

$$\hat{G}^{(l)} \subset G \equiv W_2^1(\Omega; \Gamma_0) \tag{4.3}$$

consisting of continuous on the domain Ω and piecewise linear functions (with respect to this triangulation) which vanish on Γ_0, $l = 0, \ldots, p$.

Let $\Omega^{(l)}$ be a set of vertices $P_i^{(l)}$ of triangles T_l, which do not belong to Γ_0, and let each vertex (node) $P_i^{(l)}$ be in correspondence with the standard basis continuous on Ω piecewise linear function $\hat{\psi}_i^{(l)}(x)$ such that $\hat{\psi}_i^{(l)}(P_i^{(l)}) = 1$ and $\hat{\psi}_i^{(l)} = 0$ at the remaining nodes, and $\hat{\psi}_i^{(l)}(x)$ is linear on each triangle $T_l \in T^{(l)}(\bar{\Omega})$. Then

$$\hat{G}^{(l)} \equiv \{\hat{u} : \hat{u} = \sum_{P_i^{(l)} \in \Omega^{(l)}} u_i \hat{\psi}_i^{(l)}(x)\}, \quad l = 0, \ldots, p, \tag{4.4}$$

where N_{l+1} is the number of nodes in $\Omega^{(l+1)}$, $N_{l+1} = N_l + N_l^{(1)}$,

$$\mathbf{R}^{N_{l+1}} \equiv H^{(l+1)} = H_1^{(l+1)} \times H_2^{(l+1)}, \quad H_2^{(l+1)} = H^{(l)},$$

and

$$\mathbf{u}_{l+1} = \{u_i\} \in H^{(l+1)}, \quad \mathbf{u}_{l+1} = [\mathbf{u}_1^{(l+1)}, \mathbf{u}_2^{(l+1)}]^T, \quad \mathbf{u}_s^{(l+1)} \in H_s^{(l+1)}, s = 1, 2.$$

Along with the basis $\{\hat{\psi}_i^{(l+1)}(x)\}$ for $\hat{G}^{(l+1)}, l = 0, \ldots, p-1$, we consider the hierarchical basis leading to the splitting

$$\hat{G}^{(l+1)} = \hat{G}_1^{(l+1)} \oplus \hat{G}_2^{(l+1)} \subset G, \quad l = 1, \ldots, p-1 \tag{4.5}$$

where

$$\hat{G}_2^{(l+1)} = \hat{G}^{(l)}$$

and

$$\hat{G}_1^{(l+1)} \equiv \{\hat{u} : \hat{u} \in \hat{G}^{(l+1)}, \hat{u}(P_i^{(l)}) = 0 \text{ for all } P_i^{(l)} \in \Omega^{(l)}\}. \tag{4.6}$$

Along with this splitting for $\hat{G}^{(l+1)}$, we consider

$$\vec{G}^{(l+1)} = \vec{G}_1^{(l+1)} \oplus \vec{G}_2^{(l+1)} \subset \vec{V}, \quad l \in [0, p-1], \tag{4.7}$$

where the components of the vector-functions

$$\vec{u}^{(l+1)} \equiv [\hat{u}^{(1,l+1)}, \hat{u}^{(2,l+1)}] \in \vec{G}^{(l+1)} \tag{4.8}$$

belong to the spaces $\hat{G}_1^{(l+1)}$ and $\hat{G}_2^{(l+1)}$, respectively. We emphasize that $\vec{G}_2^{(l+1)} = \vec{G}^{(l)}$ and that the components of $\vec{u}^{(l+1)} \in \vec{G}_1^{(l+1)}$ vanish at the vertices of triangles $T_k \in T^{(l)}$. We note also that the Gram matrices for the two indicated bases for the space $\vec{G}_1^{(l+1)}$ take the standard block form

$$L^{(l+1)} \equiv \begin{bmatrix} L_{1,1}^{(l+1)} & L_{1,2}^{(l+1)} \\ L_{2,1}^{(l+1)} & L_{2,2}^{(l+1)} \end{bmatrix}, \quad \bar{L}^{(l+1)} \equiv \begin{bmatrix} \bar{L}_{1,1}^{(l+1)} & \bar{L}_{1,2}^{(l+1)} \\ \bar{L}_{2,1}^{(l+1)} & L_{2,2}^{(l)} \end{bmatrix}. \tag{4.9}$$

Lemma 4.1. *The angle α between the subspaces $\vec{G}_2^{(l+1)} = \vec{G}^{(l)}$ and $\vec{G}_1^{(l+1)}$ is not smaller than the angle between the respective subspaces when $S = \emptyset$.*

Proof. It suffices to introduce the semiinner product

$$(\vec{u}, \vec{v})_S \equiv \sum_{r=1}^{m} [c_{r,1}(1, I_{r,s}(\vec{u}) I_{r,s}(\vec{v}))_{1,S_r} + c_{r,2}(1, I_{r,n}(\vec{u}) I_{r,n}(\vec{v}))_{1,S_r}] \tag{4.10}$$

and to observe that $(\vec{u}^{(l+1)}, \vec{u}^{(l)})_S = 0$ if $\vec{u}^{(l+1)} \in \vec{G}_1^{(l+1)}$. □

Note that if we deal only with isosceles rectangular triangles in $T_h(\bar{\Omega})$, then $\alpha \geq \pi/4$.

Now in accordance with the theory of optimal model operators given in [9,15–17], we need to approximate the block $\bar{L}_{1,1}^{(l+1)} = L_{1,1}^{(l+1)} \in \mathcal{L}(\vec{H}_1^{(l+1)})$ (here, $\mathcal{L}(\vec{H}_1^{(l+1)})$ refers to the space of linear operators that map $\vec{H}_1^{(l+1)}$ into itself).

Lemma 4.2. *Suppose that the basis functions for* $G_1^{(l+1)}$ *are indexed so that the two basis functions associated with each node on* S *have consecutive numbers. Then there exists a block diagonal matrix* $A_{1,1}^{(l+1)} \in \mathcal{L}(\vec{H}_1^{(l+1)})$, *with blocks in* $\mathbf{R}^{2\times 2}$ *or* $\mathbf{R}^{1\times 1}$ *and constants* $\sigma_{0,1} > 0$ *and* $\sigma_{1,1} > 0$, *independent of* l *and coefficients* $c_{r,1}$ *and* $c_{r,2}$ $(r \in [1,m])$, *such that*

$$\sigma_{0,1} A_{1,1}^{(l+1)} \le L_{1,1}^{(l+1)} \le \sigma_{1,1} A_{1,1}^{(l+1)}, \quad l+1 \in [1,p]. \tag{4.11}$$

Proof. We may take

$$A_{1,1}^{(l+1)} = A_{\emptyset,1,1}^{(l+1)} + A_{S,1,1}^{(l+1)},$$

where, for all $\vec{u}^{(l+1)} \in G_1^{(l+1)}$, we have

$$(A_{\emptyset,1,1}^{(l+1)} \mathbf{u}^{(l+1)}, \mathbf{u}^{(l+1)})_{\vec{H}_1^{(l+1)}} = |\vec{u}^{(l+1)}|_{1,\Omega}^2$$

and

$$(A_{S,1,1}^{(l+1)} \mathbf{u}^{(l+1)}, \mathbf{u}^{(l+1)})_{\vec{H}_1^{(l+1)}} = \|\vec{u}^{(l+1)}\|_S^2.$$

Moreover, we see that $A_{\emptyset,1,1}^{(l+1)}$ is a positive diagonal matrix (its elements are uniformly bounded) and $A_{S,1,1}^{(l+1)}$ is a nonnegative block diagonal matrix (its elements are $O(1/h^{(l+1)})$).□

Note that if $c_{r,1} = c_{r,2}$, $(r \in [1,m])$, then $A_{S,1,1}^{(l+1)}$ is diagonal.

Theorem 4.1. *Let the operator* $\Lambda_{1,1}$ *be the Gram matrix for the basis functions in* $\hat{G}_1$. *Then there exists an asymptotically optimal model operator* $B_1 \sim \Lambda$ *such that the constants of spectral equivalence and the estimates of the required computational work in solving systems with* B_1 *are independent of* $c_{r,1}$ *and* $c_{r,2}$ $(r \in [1,m])$.

Proof. It suffices to apply construction of the model cooperative operators $\bar{B}^{(l+1)}$ and $B^{(l+1)}$ from [9,15–17], in combination with the above lemmas. □

Now we define

$$B \equiv \begin{bmatrix} B_1 & 0 \\ 0 & B_2 \end{bmatrix}, \tag{4.12}$$

where B_2 is, for example, a diagonal matrix whose diagonal elements are areas of our triangles in $T_h(\bar{\Omega})$. Then it can be proved (see [6,9]) that

$$\|L\|_{H(B)\mapsto H(B^{-1})} \le \delta_1^{1/2}, \quad \|L^{-1}\|_{H(B^{-1})\mapsto H(B)} \le \delta_0^{-1/2},$$

where the constants are independent of h. These inequalities, probably for the first time, were discussed in [18,19] and written in the form

$$\delta_0 \|v\|_B^2 \le \|Lv\|_{B^{-1}}^2 \le \delta_1 \|v\|_B^2, \quad \forall v \in H.$$

They may be regarded as a consequence of the correctness of the original elliptic problem and they yield inequalities

$$\delta_0 B \le L^* B^{-1} L \le \delta_1 B$$

and

$$\text{sp } (B^{-1} L^* B^{-1} L) \subset [\delta_0, \delta_1]$$

(see [6,9,19,20]). Therefore, for solving (4.1), it is reasonable to use iterations

$$Bu^{n+1} = Bu^n - \tau_n (L^* B^{-1} (Lu^n - f), \tag{4.13}$$

convergence of which is determined by the constants δ_0 and δ_1 (more precisely, by their quotient). Note that if we consider L as mapping of the Euclidean space $H(B)$ into $H(B^{-1})$ then its conjugate operator L' is given by $L' = B^{-1} L^* B^{-1}$ (in our case, we have $L^* = L$).

Thus, we actually work with the symmetrization defined by the chosen pair of spaces; it leads to the system

$$Au \equiv B^{-1} L^* B^{-1} Lu = B^{-1} L^* B^{-1} f$$

with the symmetric operator A considered as a mapping of the Euclidean space $H(B)$ into itself.

In case of the modified Richardson method (4.13), the adaptation procedure from [6,9] for the constants δ_0 and δ_1 is available; the modified conjugate gradient method can also be used.

5. EXAMPLE OF SPECTRAL PROBLEMS

Next, consider the special case of spectral problems in the strengthened Sobolev spaces (this problem is connected with estimating linear interpolation error in a triangle for a function with standard extra smoothness and is important for error estimates of the finite element method associated with piecewise linear functions; similar problems were considered in [3,18] for common Sobolev spaces).

Let T be the triangle with vertices $(0,0)$, $(1,0)$, and $(0,1)$, and let the strengthened Sobolev space W consist of functions $w \in W_2^2(T)$ that vanish at these vertices and such that

$$\|D_2^2 w\|_{0,S} < \infty,$$

where S denotes the vertical side of T. In other words, we assume that

$$\int_0^1 (D_2^2 w)^2 dx_2 < \infty.$$

We define the inner product in this strengthened Sobolev space by

$$(w, w')_W \equiv (w, w')_{2,T} + (D_2^2 w, D_2^2 w')_{0,S}. \tag{5.1}$$

We seek

$$\lambda_1 \equiv \min_{w \in W \setminus 0} \frac{|w|_W^2}{|w|_{1,T}^2 + |D_2 w|_{0,S}^2} \tag{5.2}$$

(see (3.2), (3.3) with $\gamma = 1$).

The Hilbert space G_1 consists now of vector functions

$$u_1 \equiv \vec{u}_1 \equiv [u_{1,1}, u_{1,2}] \in (W_2^1(T))^2$$

such that

$$\phi_1(\vec{u}_1) \equiv \int_0^1 u_{1,1}(0, x_2) dx_2 = 0, \quad \phi_2(\vec{u}_1) \equiv \int_0^1 u_{1,2}(x_1, 0) dx_1 = 0, \tag{5.3}$$

and

$$|D_2 u_{1,1}|_{0,S} < \infty. \tag{5.4}$$

We define the inner product in this strengthened Sobolev space by

$$(\vec{u}_1, \vec{u}_1')_{G_1} \equiv (\vec{u}_1, \vec{u}_1')_{1,T} + (D_2 u_{1,1}, D_2 u_{1,1}')_{0,S}. \tag{5.5}$$

The space rot W is defined by

$$\text{rot } W \equiv \{u_1 : u_1 \in G_1 \text{ and div } u_1 = 0\}. \tag{5.6}$$

Theorem 5.1. *Problem* (5.2) *is equivalent to finding*

$$\lambda_1 \equiv \min_{u_1 \in rot\, W \setminus 0} \frac{|u_1|_{G_1}^2}{|u_1|_{0,T}^2 + |u_{1,1}|_{0,S}^2} \tag{5.7}$$

and is reduced to a particular case of the eigenvalue problem:

$$(u_1; v_1)_{G_1} + b(v_1; u_2) = \lambda[(u_1; v_1)_{0,T} + (u_{1,1}, v_{1,1})_{0,S}], \quad \forall v_1, \tag{5.8}$$

$$b(u_1; v_2) = 0, \quad \forall v_2, \tag{5.9}$$

where

$$b(u_1; v_2) = (\text{div } u_1, v_2)_{0,T}. \tag{5.10}$$

Proof. It suffices to rewrite (5.2) in terms of rot w. It is also simple to obtain (2.6) for the indicated G_1 and $G_2 \equiv L_2(T)$. □

It is not difficult to show that, for the indicated problem, we may apply PGMs, based on quasiuniform triangulations and local refinements around the vertices of T and in the vicinity of S. We stress that the basis for the approximating spline subspaces $\hat{G}_{1,h}$ and $\hat{G}_{2,h}$ are the same as for the case $S = \emptyset$ (elements of $\hat{G}_{1,h}$ must satisfy nonlocal condition (5.3)). We thus obtain PGM of the form: Find $u_1 \in \hat{G}_{1,h}$ and $u_1 \in \hat{G}_{2,h}$ such that $u_1 \neq 0$ and

$$(u_1; v_1)_{G_1} + b(v_1; u_2) = \lambda(u_1; v_1)_{0,}, \quad \forall v_1 \in \hat{G}_{1,h}, \tag{5.11}$$

$$b(u_1; v_2) = 0, \quad \forall v_2 \in \hat{G}_{2,h}. \tag{5.12}$$

It is very important that (3.1) holds.

Problem (5.11), (5.12) can be rewritten in operator form (in the Euclidean space $H \equiv H_1 \times H_2$) as

$$Lu \equiv \begin{bmatrix} L_{1,1} & L_{1,2} \\ L_{2,1} & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \lambda M u \equiv \lambda \begin{bmatrix} M_1 u_1 \\ 0 \end{bmatrix}. \tag{5.13}$$

To obtain effective algorithms for (5.13), we suggest the penalty method, yielding the problem

$$\begin{bmatrix} L_{1,1} & L_{1,2} \\ L_{2,1} & -\alpha J_2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \lambda \begin{bmatrix} M_1 u_1 \\ 0 \end{bmatrix}, \quad \alpha > 0, \tag{5.14}$$

where J_2 is a diagonal positive matrix with the diagonal elements equal to areas of augmented triangles in $T_h(T)$. Problem (5.14) is reduced to the standard problem

$$S_1 u_1 \equiv (L_{1,1} + 1/\alpha L_{1,2} J_2^{-1} L_{2,1}) u_1 = \lambda M_1 u_1 \tag{5.15}$$

in the Euclidean space H_1. Moreover, an asymptotically optimal model operator B_1 for $L_{1,1}$ can be constructed like in Section 4 (L_1 and B_1 are spectrally equivalent operators). This implies that it is possible to indicate a nearly asymptotically optimal model operator D_1 for S_1 with the constants of spectral equivalence independent of α (operators of such type were obtained in [4,6]). Therefore, it is possible to indicate nearly asymptotically optimal algorithms for problems of type (5.8)–(5.10) and solve them with high accuracy if need be. But, when a moderate accuracy is required, a simpler approach based on the penalty method might be more useful. For example, as in [21], it is possible to replace (5.7) by

$$\lambda_1(\alpha) \equiv \min_{u_1 \in G_1 \setminus 0} \frac{|u_1|^2_{G_1} + 1/\alpha |\mathrm{div}\, u_1|^2_{0,T}}{|u_1|^2_{0,T} + |u_{1,1}|^2_{0,S}} \tag{5.16}$$

and make use of the simplest triangulation and PGM.

Note that along the same lines we can consider problems that have stiffeners on other sides of our triangle.

Our approach can be generalized to more general spectral problems in the strengthened Sobolev spaces typical for stability analysis of stiffened plates.

REFERENCES

[1] S. C. Brenner. An optimal-order nonconforming multigrid method for the biharmonic equation, *SIAM J. Numer. Anal.*, **26**, 1124–1138, 1989.

[2] S. C. Brenner. A nonconforming mixed multigrid method for the stationary Stokes equations, *Math. Comp.*, **55**, 411–437, 1990.

[3] E. G. D'yakonov. Methods of solving fourth order elliptic problems that are asymptotically optimal with respect to labor consumption, *Soviet Math. Dokl.*, **32**, 128–132, 1985.

[4] E. G. D'yakonov. Effective methods for solving eigenvalue problems with fourth-order elliptic operators, *Soviet J. Numer. Anal. Math. Modelling*, **1**, 59–82, 1986.

[5] E. G. D'yakonov. On iterative methods with saddle operators, *Soviet Math. Dokl.*, **35**, 166–170, 1987.

[6] E. G. D'yakonov. *Minimization of Computational Work. Asymptotically Optimal Algorithms for Elliptic Problems*, Nauka, Moscow, 1989 (in Russian).

[7] E. G. D'yakonov. On some iterative methods for nonlinear grid systems, in *Computational Processes and Systems, 8*, Marchuk, G. I., Ed., Nauka, Moscow, 95–111, 1991 (in Russian).

[8] E. G. D'yakonov. Composite grids and asymptotically optimal algorithms for problems of Stokes and Navier–Stokes type, *Russian Acad. Sci. Dokl. Math.*, **47** , 221–227, 1993.

[9] E. G. D'yakonov. *Optimization in Solving Elliptic Problems*, CRC Press, Boca Raton, 1995.

[10] R. Courant. Variational methods for the solution of problems of equilibrium and vibrations, *Bull. of Amer. Math. Soc.*, **49**, 1–23, 1943.

[11] M. Krizek, P. Neittaanmaki, and R. Stenberg, Eds. *Finite Element Methods*, Marcel Dekker, Inc., New York, 1994.

[12] M. D. Gunzburger. *Finite Element Methods for Viscous Incompressible Flow: A Guide to Theory, Practice and Algorithms*, Academic Press, Boston, 1989.

[13] R. Stenberg. Error analysis of some finite element methods for the Stokes problem, *Math. Comp.*, **54**, 495–508, 1990.

[14] Eugene et Francois Cosserat. Sur les equations de la theorie de l'elastite, *C. R. Acad. Sci. (Paris)*, **126**, 1089–1091, 1898.

[15] O. Axelson and P. S. Vassilevski. A survey of multilevel preconditioned iterative methods, *BIT*, **29**, 769–793, 1989.

[16] E. G. D'yakonov. On some modern approaches to constructing spectrally equivalent grid operators, in *Fourth International Symposium on Domain Decomposition Methods for Partial Differential Equations,* Glowinski, R., Kuznetsov, Yu. A., Meurant, G., Periaux, J., and Widlund, O. B., Ed.s, SIAM, Philadelphia, 35–40, 1991.

[17] E. G. D'yakonov. On increasing the efficiency of grid methods for solution of elasticity problems, in *Computer Mechanics of Solids*, Moscow, n. 2, 133–157, 1991 (in Russian).

[18] E. G. D'yakonov. The use of spectrally equivalent operators in solving difference analogs of strongly elliptic systems, *Soviet Math. Dokl.*, **6**, 1105–1109, 1965.

[19] E. G. D'yakonov. The construction of iterative methods based on the use of spectrally equivalent operators, *USSR Comp. Math. and Math. Phys.*, **6**, 14–46, 1966.

[20] E. G. D'yakonov. Iterative methods based on linearization for nonlinear elliptic grid systems, in *Numerical Methods and Applications*, Marchuk, G. I., Ed., CRC Press, Boca Raton, 1–43, 1994.

[21] G. L. Siganevich. On the best error estimate of the linear interpolation on a triangle for functions in $W_2^2(T)$, *Soviet Math. Dokl.*, **37**, 745–748, 1988.

A Parallel Multilevel Spectral Element Scheme

M.B. Davis and G.F. Carey
ASE-EM Dept., The University of Texas at Austin

March 5, 1996

Abstract

A parallel multilevel strategy is developed using spectral (p) finite elements. Hierarchic bases are particularly well suited since the element matrices and vectors are nested and the multilevel projections easily performed. Since the basis degree is used to specify the multigrid level, an EBE strategy is natural for the multilevel technique. Results are presented for two candidate nonlinear elliptic transport problems: the augmented drift-diffusion equations of semiconductor device modeling and the stream function-vorticity equations of incompressible fluid dynamics.

Introduction

Finite element methods in which refinement is accomplished by increasing the degree p of the polynomial basis can give superior error convergence rates for similar computational work than the more commonly used h refinement schemes. However, the condition number of the matrix deteriorates with increasing p. This motivates the need for an effective preconditioner, and a multilevel scheme in which the basis degree serves as the grid level is a natural choice. Hierarchic basis functions, which are constructed by adding appropriate functions to the existing lower-degree polynomials, lead to matrices and vectors which are nested. This may be a particularly suitable choice for multilevel methods, since the projections for hierarchic multilevel schemes are easily performed at little computational cost [8, 9].

Element-by-element strategies have proven to be efficient and scalable for parallelization of finite element methods using gradient iterative solvers [1, 2, 5, 6]. The basic idea in the parallel EBE scheme is to avoid assembling the system and instead perform matrix-vector and dot products in parallel at the element level. All matrices and vectors are stored in element format. Moreover, in this approach, multilevel operations such as residual calculation, restriction and prolongation can be confined to an element and hence require no interprocessor communication. The only steps that require communication are the smoothing iterations and the coarse level solve. A further advantage of spectral multilevel methods is that the number of elements in the domain remains constant, and hence the decomposition of the domain is fixed across grid levels. An important issue with parallel multilevel methods defined in this way is the ratio of communication to calculation. Although this ratio may be small for the fine level (high-degree basis), on coarser levels it gets successively

larger, and at some point the communication time may dominate the total computational time. Further details of the p-type approach are given in [7, 8].

P-type Multilevel Scheme

An alternative to refining the mesh by making the element size h smaller is to increase the degree p of the polynomial basis. This strategy results in exponential convergence when the solution is sufficiently regular. One disadvantage of the p-type finite element method, however, is that the conditioning of the matrix deteriorates with increasing p. This deterioration is dependent on the type of basis used. One way to counter this is to apply a preconditioner to the system. A p-type multilevel method may be defined by using the degree of the polynomial basis as the grid level. The intergrid transfers can then be naturally defined in terms of expansions in the appropriate bases.

The analysis of a finite element Galerkin multilevel scheme is best carried out in the variational setting. In this way the Galerkin statement can be formulated on each grid level, and the consistency of the projection operators with the finite element discretizations on the associated grid levels is assured. The approach here follows that in [4, 8]. We proceed by considering a representative linear elliptic problem on a domain Ω with a boundary $\partial\Omega$:

$$L(u) = f \text{ in } \Omega \tag{1}$$

$$u = g \text{ on } \partial\Omega \tag{2}$$

where L denotes the differential operator. Applying the method of weighted residuals and integrating by parts, the variational statement of the problem has the form: Find $u \in H$ with $u = g$ on $\partial\Omega$ such that

$$a(u, v) = f(v) \quad \forall v \in H \tag{3}$$

with $v = 0$ on $\partial\Omega$. Here $a(\cdot, \cdot)$ denotes the bilinear functional, $f(\cdot)$ is a linear functional and H is the appropriate space of admissible functions. Introducing a finite element discretization and a polynomial basis so that $S^p \subset H$, we define the approximate variational problem on grid level p as: find $u_p \in S^p$ with $u_p = g$ on $\partial\Omega$ such that

$$a(u_p, v_p) = f(v_p) \quad \forall v_p \in S_0^p \tag{4}$$

where the subscript on S^p indicates that the test functions $v_p = 0$ on $\partial\Omega_p$. Introducing the finite element expansion and evaluating the integrals in (4) leads to a linear system of the form

$$\mathbf{A}_p \mathbf{u}_p = \mathbf{b}_p \tag{5}$$

where p once again indicates the grid level. Now consider a multilevel scheme where (5) corresponds to the fine grid system. Application of an iterative smoother to this system yields an approximation u_p^* and associated error $e_p^* = u_p - u_p^*$. Substituting this into (4), the error e_p^* is specified by the residual equation

$$a(e_p^*, v_p) = r^*(v_p) \text{ for all } v_p \in S_0^p \tag{6}$$

where

$$r^*(v_p) = f(v_p) - a(u_p^*, v_p). \tag{7}$$

Next, introduce a coarser level q such that $S^q \subset S^p$. Since all v_q are in S^q and thus in S^p we can test against the set of bases v_q [4] so the solution of (6) also satisfies the property

$$a(e_p^*, v_q) = r^*(v_q) \text{ for all } v_q \in S_0^q \tag{8}$$

where $r^*(v_q) = f(v_q) - a(u_p^*, v_q)$. This system is obviously underdetermined, so we take the best (Galerkin) approximation $e_q^* \in S_0^q$ to e_p^*. That is, find $e_q^* \in S_0^q$ such that

$$a(e_q^*, v_q) = r^*(v_q) \text{ for all } v_q \in S_0^q \tag{9}$$

Substituting the finite element expansion in (9) yields the coarse level system for the error correction vector

$$\mathbf{A}_q \mathbf{e}_q^* = \mathbf{r}_q. \tag{10}$$

where $\mathbf{A}_q$ is computed by evaluating the bilinear form on the space S^q and the right side vector defines a natural projection of the residual from S^p to S^q. More specifically, (7) implies

$$r^*(v_q) = f(v_q) - a(u_p^*, v_q). \tag{11}$$

Note that this requires the $a(\cdot, \cdot)$ inner product of u_p^* and v_q.

Introducing a polynomial expansion for u_p^* and polynomial test function v_q

$$u_p^* = \sum_{j=1}^{N_p} (\beta_p^*)_j \phi_j^p(\mathbf{x}), \quad v_q = \phi_i^q(\mathbf{x}) \tag{12}$$

where ϕ_j^p and ϕ_i^q denote the respective basis functions for S^p and S^q and $(\beta_p^*)_j$ are the nodal degrees of freedom. Upon substitution in (11) this yields

$$r^*(\phi_i^q) = f(\phi_i^q) - \sum_{j=1}^{N_p} A_{ij}^{q,p} (\beta_p^*)_j \tag{13}$$

where $A_{ij}^{q,p} = a(\phi_i^q, \phi_j^p)$, or in matrix form

$$\mathbf{r}_q^* = \mathbf{f}_q - \mathbf{A}^{q,p} \beta_p^* \tag{14}$$

Now $\mathbf{r}_q^*$ in (14) can also be computed in a more traditional manner by developing a projection of $\mathbf{r}_p^*$ from the high level space to the coarser level space as follows: First, expand the test function ϕ_i^q in the higher-dimensional basis as

$$\phi_i^q = \sum_{k=1}^{N_p} m_{ik}^{q,p} \phi_k^p \tag{15}$$

Then, substituting (15) into (13),

$$\begin{aligned} r^*(\phi_i^q) = &= f(\phi_i^q) - a(u_p^*, \phi_i^q) \\ &= \sum_{j=1}^{N_p} m_{ij}^{p,q} f(\phi_j^p) - \sum_{j=1}^{N_p} m_{ij}^{q,p} a(u_p^*, \phi_j^p) \\ &= \sum_{j=1}^{N_p} m_{ij}^{q,p} r^*(\phi_j^p) \end{aligned} \tag{16}$$

or in matrix form

$$\mathbf{r}_q^* = \mathbf{M}^{q,p}(\mathbf{b}_p - \mathbf{A}_p\mathbf{u}_p^*) = \mathbf{M}^{q,p}\mathbf{r}_p^* \tag{17}$$

At this point we need to determine the actual values of $m_{ik}^{q,p}$ in order to be able to carry out the projection. First let us consider the standard Lagrange bases. These bases have the interpolation property: the value of each basis function is one at the node corresponding to the basis function, and zero at all other nodes, *i.e.* $\phi_i(\mathbf{x}_j) = \delta_{ij}$. It follows that

$$\phi_i^q(\mathbf{x}_j) = \sum_{k=1}^{N_p} m_{ik}^{q,p}\phi_k^p(\mathbf{x}_j) = m_{ij}^{q,p} \tag{18}$$

and the components of the projection matrix $\mathbf{M}^{q,p}$ are simply the values of the coarse grid basis at the fine grid nodes.

To complete the multilevel concept in the variational setting, a prolongation operator is needed which will project the error correction in equation (10) to grid level p. A natural choice for the prolongation operator is the transpose of the restriction operator in (17). Then the fine grid correction approximating e_p^* in S^p is computed from the coarse grid result according to

$$\tilde{\mathbf{e}}_p = (\mathbf{M}^{q,p})^T\mathbf{e}_q^* \tag{19}$$

As in the standard multigrid method, these error corrections are added to the approximate solution on the finer level to obtain the corrected approximation $\tilde{\mathbf{u}_p} = \mathbf{u}_p^* + \tilde{\mathbf{e}}_p$ and smoothed by fine grid iteration to get a new $\mathbf{u}_p^*$ for the next V-cycle.

The advantages of hierarchic bases become apparent when we extend the previous multilevel analysis to this setting [7, 8]. The change-of-basis coefficients in (15) for Lagrange bases are simplified for hierarchics because the basis for the space S^q is explicitly contained in the basis for S^p. That is,

$$\phi_i^q = \phi_i^p \quad 1 \le i \le N_q \tag{20}$$

which implies

$$m_{ij}^{q,p} = \delta_{ij} \qquad i = 1, \ldots, N_q, \quad j = 1, \ldots, N_p. \tag{21}$$

Since the higher-degree basis explicitly contains the lower-degree basis, the finite element matrix and vector contributions corresponding to the lower-degree polynomials are nested in the matrix and vector contributions for the higher-degree polynomials. Similarly, coarsening implies simply deleting the appropriate rows and columns of the matrix. These properties are useful in the multilevel context. More specifically, the residual projection in (13) becomes $r^*(\phi_i^q) = r^*(\phi_i^p)$ for $i = 1, 2, \ldots, N_q$. That is, the components of the residual projection to the subspace S^q are trivially the first N_q components of the fine grid residual. Hence, only the first N_q components of the residual vector need to be computed. Similarly, the coarse grid matrix $\mathbf{A}_q$ is now the leading $N_q \times N_q$ minor of the fine level matrix $\mathbf{A}_p$. Hence $\mathbf{A}_q$ does not need to be recomputed.

The subspace problem for the error correction in S^q again has the form in (10). That is,

$$\mathbf{A}_q\mathbf{e}_q^* = \mathbf{r}_q^*. \tag{22}$$

In a two-level scheme this system is solved for $\mathbf{e}_q^*$. The projection of $\mathbf{e}_q^*$ to the higher level space S^p is trivial because of the explicit inclusion of the basis (recall (20)). Hence the

corrected high level approximation is simply obtained by adding the N_q components of $\mathbf{e}_q^*$ to the first N_q components of $\mathbf{u}_p^*$. This new approximation in S^p can then be iteratively smoothed and the cycle repeated.

Since $\mathbf{M}^{q,p}$ extracts the first N_q components of a vector of length N_p, equation (22) on the coarse grid can then be expressed as

$$\mathbf{A}_q\mathbf{e}_q^* = \mathbf{M}^{q,p}(\mathbf{b}_p - \mathbf{A}_p\mathbf{u}_p^*) = \mathbf{b}_q - \mathbf{M}^{q,p}\mathbf{A}_p\mathbf{u}_p^* \tag{23}$$

An alternative to the standard error correction method described above takes advantage of the nesting of the matrices and vectors [9] to operate directly on the associated components of $\mathbf{u}_p^*$ and $\mathbf{b}_p$. First note that $\mathbf{M}^{q,p} = [\mathbf{I}\ \ \mathbf{0}]$ so (23) implies

$$\mathbf{A}_q\mathbf{e}_q^* = \mathbf{b}_q - [\mathbf{A}_q\ \ \mathbf{A}_{qp}]\mathbf{u}_p^* = \mathbf{b}_q - \mathbf{A}_q\mathbf{u}_q^* - \mathbf{A}_{qp}\mathbf{u}_{pp}^* \tag{24}$$

where we have used the block partitioning

$$\mathbf{A}_p = \begin{bmatrix} \mathbf{A}_q & \mathbf{A}_{qp} \\ \mathbf{A}_{pq} & \mathbf{A}_{pp} \end{bmatrix} \quad \mathbf{u}_p^* = \begin{bmatrix} \mathbf{u}_q^* \\ \mathbf{u}_{pp}^* \end{bmatrix} \tag{25}$$

then (24) implies, on transposing $\mathbf{A}_q\mathbf{u}_q^*$

$$\mathbf{A}_q\tilde{\mathbf{u}}_q = \mathbf{b}_q - \mathbf{A}_{qp}\mathbf{u}_{pp}^*. \tag{26}$$

where $\tilde{\mathbf{u}}_q = \mathbf{e}_q^* + \mathbf{u}_q^*$ is the subvector corresponding to the first N_q components of the new high-level iterate. This form has two advantages. First, it emphasizes the fact that the full residual need not be computed. Second, no intermediate correction needs to be projected and added to the fine level approximation.

For reasons of convenience and parallelization, a simple point Jacobi iteration is the preferred smoother for the multilevel scheme. Any smoother must efficiently damp the high frequency error modes on the respective grids. For the relaxed Jacobi smoother, the relaxation parameter determines which frequencies are damped more quickly than others. If we assume that we wish to eliminate the highest frequency eigenmode corresponding to the leading eigenvalue of the discrete operator, we obtain the relaxation factor for optimum multilevel smoothing [7, 15]

$$\omega = \left(\frac{(\mathbf{x}, \mathbf{A}\mathbf{x})}{(\mathbf{x}, \mathbf{D}\mathbf{x})}\right)^{-1} \tag{27}$$

where $\mathbf{D}$ is a diagonal matrix with $D_{ii} = A_{ii}$.

Since this relaxation factor ω is a function of the matrix $\mathbf{A}$, it changes with both the problem and the discretization. Hence, the optimum relaxation needs to be repeatedly calculated for each decoupled equation matrix. This value can be conveniently calculated using a power series method.

There are two main choices for a multilevel strategy applied to a linear system: a V (or W) cycle, or a full multigrid cycle. The full multigrid (FMV) cycle uses nested iteration to improve the initial guess on successively finer grids. The strategy for solution of the nonlinear problem uses block iteration and successive approximations. Hence, at each nonlinear (or block) iteration, there exists a good initial guess on the fine grid. For this reason only V-cycles are used here as a multigrid cycling scheme.

The Jacobi smoother can generate oscillations in the cross-wind direction for convection dominated problems. The magnitude of these oscillations is proportional to the magnitude of the residual. In order to minimize these oscillations, an initial coarse grid correction (no pre-smooths) is performed at the first V-cycle. This initial correction further improves the initial guess from the previous block iterate, and convergence is improved [7].

Multigrid cycling schemes such as the Full Approximation Scheme can be used on the full nonlinear problem. Two alternative approaches are used here for the nonlinear problem. First, the multilevel solver is used only as the linear system solver for the fine grid problem, which is run to convergence using successive approximations and continuation. The second approach is a nested iteration scheme: The coarsest grid problem is run to convergence on the full nonlinear problem, including continuation in the boundary voltage or Reynolds number. The solution is then projected to the next finest grid and the problem on this grid level is then run to convergence at the final voltage or Reynolds number. This strategy is repeated until the highest grid level is reached.

Parallelization

Finite element methods divide a given problem domain into a union of elements for discrete solution. Hence, schemes in which blocks of elements are operated on by a processor and the processor decomposition follows element boundaries provide a natural way in which to parallelize finite element methods [1, 2, 5, 6]. Adjacent elements share nodes on the element interface, so the information associated with these nodes may be stored on different processors. This information is updated during matrix-vector product or inner product operations. This means that messages must be passed between processors in order to update these values. The ratio of communication to computation is important because it can limit efficiency. The use of high-p elements, which have more internal degrees of freedom, results in a higher computation to communication ratio compared to low-p elements.

For a message passing paradigm, the time to send a message is given by

$$t_m = \alpha + \beta L_m \tag{28}$$

where α is the startup time or latency, β is the time per byte for message transfer, and L_m is the length of the message in bytes. For transfers in which a large amount of data is to be transferred, the key is to send as few messages as possible so that the startup time is minimized. Otherwise the startup time may dominate the communication time. The optimum situation would be to send one long message so that the latency is essentially hidden.

The previous argument motivates the need for message bundling using sendlists. A data structure is developed in which each processor has a pointer array which contains the element and node numbers that are shared with another processor. The order in which this information is to be placed into a message is also stored. Thus, when a vector is to be updated, a message vector is filled in order and sent to the appropriate processor. In turn, a message is received from that processor. A pointer array indicates which element and local node corresponds to which position in the array, in the same way as for the message which was sent. In this fashion all of the communication between adjacent processors can

be accomplished using one message each way, and message latency is minimized. There is, however, some overhead in the packing and unpacking phases.

In the present work we can use an element-type data structure and recast all matrix-vector or projection operations at the element level. This means that instead of addressing a vector by its global node number, it is addressed by its element and local node number. In addition, each element has a pointer array which stores its neighbor elements and which points are shared with this neighbor. A specific processor will store information only for elements local to that processor. Elements are therefore addressed by the number local to that processor rather than a global element number. The pointer array for neighbor information includes the local element number and processor number for neighboring elements. This format facilitates parallel coding.

The formation of the matrix and RHS vector for finite element methods is usually accomplished by forming the local element matrices and vectors and summing them to get the global matrix and RHS as implied in the multilevel formulation of the previous sections. However, in the present parallel algorithm we no longer form the global matrix and RHS, but leave them in element form. The matrix and RHS calculation phase is therefore completely parallel. If the matrix is to be preconditioned using a global Jacobi preconditioner (diagonal scaling), then the diagonal elements of the matrices may be assembled to find the scaling factor. This accumulation phase will involve communication across processor boundaries.

Iteration by point iterative methods (Jacobi, SOR, etc.) as a smoother or gradient methods (CG, BCG, etc.) for the coarse grid solve involves repeated matrix-vector multiplications or dot products. Calculation of either one requires that the information on shared nodes be updated. However, the multilevel residual calculation and projection operations require no communication. The residual calculation on the fine grid is (17)

$$\mathbf{r}_p = \mathbf{b}_p - \mathbf{A}_p \mathbf{u}_p^* \tag{29}$$

This is seen, however, as a sum of element contributions. Let $\mathbf{A}_p^e, \mathbf{r}_p^e, \mathbf{b}_p^e$ be the element matrix and vectors, respectively. Introducing the Boolean adjacency or connectivity matrix which relates global to local variables for element e

$$\mathbf{b}_p = \sum_{e=1}^{E} \mathbf{B}_e^T \mathbf{b}_p^e \tag{30}$$

and similarly

$$\mathbf{A}_p = \sum_{e=1}^{E} \mathbf{B}_e^T \mathbf{A}_p^e \mathbf{B}_e \tag{31}$$

Then

$$\begin{aligned} \mathbf{r}_p &= \sum_{e=1}^{E} \mathbf{B}_e^T \mathbf{b}_p^e - \sum_{e=1}^{E} \mathbf{B}_e^T \mathbf{A}_p^e \mathbf{B}_e \mathbf{u}_p^* \\ &= \sum_{e=1}^{E} \mathbf{B}_e^T (\mathbf{b}_p^e - \mathbf{A}_p^e \mathbf{u}_p^e) \\ &= \sum_{e=1}^{E} \mathbf{B}_e^T \mathbf{r}_p^e \end{aligned} \tag{32}$$

and we can use directly the element residuals

$$\mathbf{r}_p^e = \mathbf{b}_p^e - \mathbf{A}_p^e \mathbf{u}_p^e \tag{33}$$

Note also that because the element bases are defined locally we can introduce a local change of basis at the element level and corresponding to the global matrix $\mathbf{M}^{q,p}$ in (17) or (21) we have the element projection matrix $\mathbf{M}_e^{q,p}$. Then the element residual projection follows in a manner analogous to (17) as

$$\mathbf{r}_q^e = \mathbf{M}_e^{q,p} \mathbf{r}_p^e \tag{34}$$

Thus residual calculation and restriction take place on the element level, without communication, and are completely parallel operations. The prolongation to finer grid operates on the error vector, which is the solution on the coarser grid. This vector is stored in summed format, and hence no updating is necessary. Therefore, prolongation can also take place on an element and is once again completely parallel.

Results

The above method is now formulated for two nonlinear, coupled transport problems. The first test case is the augmented drift-diffusion equations, which model the flow of electrons and holes in semiconductor devices. The steady state, scaled form of the equations is [3, 12, 16, 17]

$$\begin{aligned} \lambda^2 \Delta \psi &= n - p - C \\ \nabla \cdot (\mu_n \nabla n - \mu_n n \nabla \psi) &= R \\ \nabla \cdot (\mu_p \nabla p + \mu_p p \nabla \psi) &= R \end{aligned} \tag{35}$$

where ψ is the electrostatic potential, n and p are carrier concentrations, μ_n and μ_p are mobilities, R is the recombination-generation rate, C is the doping, and λ is the scaled Debye length. The boundary conditions are Dirichlet at the contacts (ψ, n, p specified) and homogeneous Neumann elsewhere.

Equations (35) are decoupled iteratively and successive approximations used to solve the nonlinear problem [7, 8, 10]. At each nonlinear iteration, three linear subsystems are obtained, which are solved successively with a multilevel method using available solution iterates of the other field variables [7].

The model problem for the augmented drift-diffusion equations is an $n^+ - n - n^+$ diode with doping of 5×10^{17} and 2×10^{15} in the n^+ and n regions, respectively, device length of $0.3\mu m$, active length of $0.1\mu m$, and an applied bias of $0.5V$. Plots of the electrostatic potential and electron concentration from source to drain contact are shown in Figure 1. Although this is a 1-D problem, it was solved on a 2-D domain with homogeneous Neumann conditions on the two horizontal sides. This solution was computed using a uniform 9×9 grid of 81 quintic elements, and a multilevel solver which used linear elements as the coarsest level.

The second application is the stream function-vorticity equations for incompressible Navier-Stokes flow in two dimensions. The steady state form of the equations is [7, 13, 14]

$$\begin{aligned} -\nu \Delta \zeta + \mathbf{u} \cdot \nabla \zeta &= f \\ -\Delta \psi &= \zeta \end{aligned} \tag{36}$$

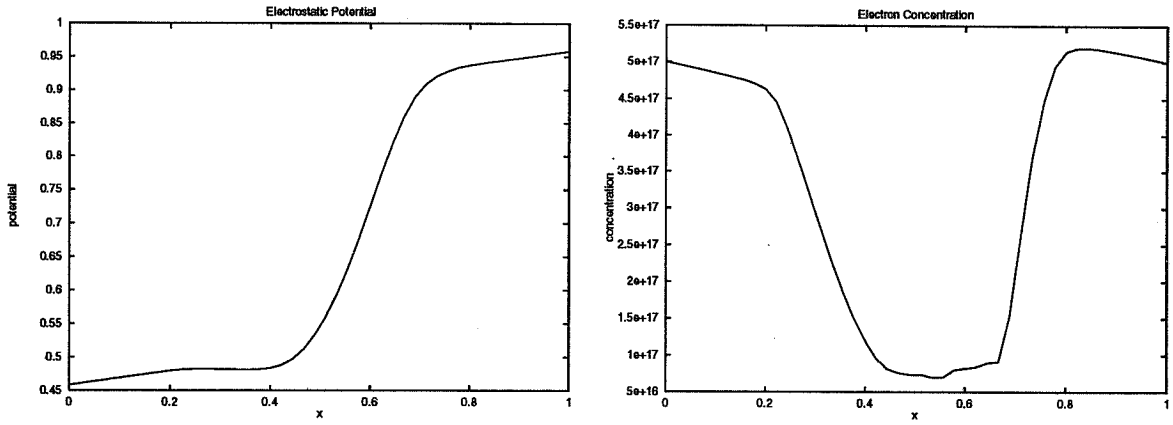


Figure 1: Potential and electron distribution solutions, $n^+ - n - n^+$ diode, $0.5V$ bias

where ψ is the stream function, ζ is the vorticity, $\mathbf{u}$ is the velocity, and f is the divergence of the body force.

Following the same procedure as in the previous problem, the equations are iteratively decoupled using successive approximations. Again, the linear systems arising from substitution of the appropriate basis and integration are solved with a multilevel scheme, and available solution iterates of the field variables are used.

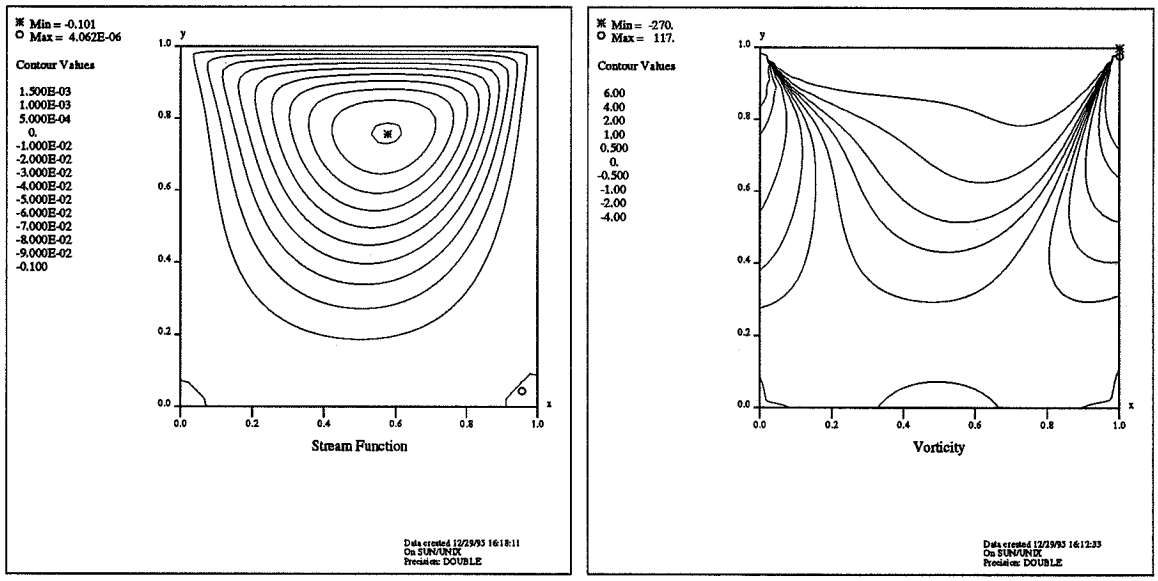


Figure 2: Stream function and vorticity contours, driven cavity, $Re = 50$

The model problem for the stream function-vorticity equations is the driven cavity problem. The velocity of the top of the cavity is normalized to one and the viscosity is chosen so that the Reynolds number of the flow is 50. Contour plots for the stream function and vorticity are shown in Figure 2. The same grid and 5-level scheme was used as in the previous case.

Calculation of the eigenvalue (or relaxation parameter) for the relaxed Jacobi smoother

in a power series scheme generally requires a moderate number of matrix-vector multiplies. If this were done at each nonlinear iteration for each decoupled linear system, the cost would quickly become a significant part of the total computation and communication time. However, if the sparse systems corresponding to a particular equation don't change enough to significantly alter this eigenvalue estimate over several block iterations, then the calculation can be done infrequently, and the cost can be amortized over several nonlinear iterations. In practice, this is found to be the case for both the augmented drift-diffusion and stream function-vorticity equations. Hence the relaxation parameter is only recomputed every ten block iterations, or at the start of a continuation step.

For multilevel schemes applied to a decoupled problem, there are two convergence rates of interest. The first is the convergence rate of the multilevel solver operating on a particular linear system. The second is the convergence rate of the successive approximation method applied to the nonlinear problem, the so-called block iterations.

Figure 3 shows the L^2 norm of the residual at each fine grid smoothing step for the augmented drift-diffusion problem, on grids of 576 quadratic elements and 81 quintic elements, respectively. These two grids have approximately the same number of degrees of freedom. The solver uses the number of levels equal to the degree of the fine grid basis. The plots display a sawtooth shape, with the beginning of each sawtooth corresponding to the formation of the new linear system at each nonlinear iteration. This is followed by a linear portion which represents the convergence of the multilevel scheme to the solution of the linear system. The large jump in each of the plots is the beginning of the new continuation step in applied voltage, which corresponds to a new problem. The linear behavior of multilevel convergence is evident and is to be expected. The convergence rate is better for the potential equation than for the transport equation. This is due to the fact that the transport equation leads to a linear system which is nonsymmetric. The envelope of the peaks is also decreasing, and this represents the convergence of the nonlinear iterations.

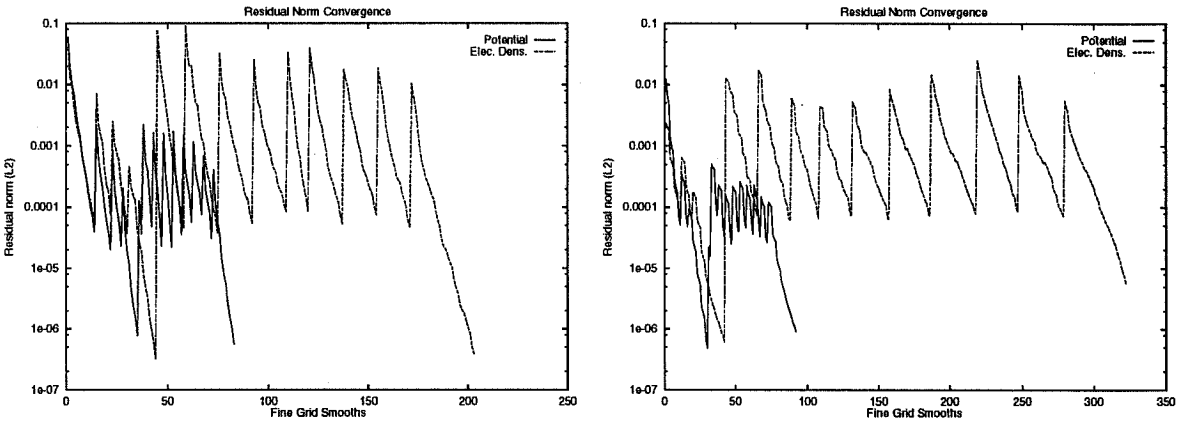


Figure 3: Multigrid convergence, augmented drift-diffusion, quadratic and quintic elements

The same types of behavior are demonstrated in Figure 4 for the stream function-vorticity problem at $Re = 50$. The convergence of the nonlinear iterations is more defined for this problem, and there is no big jump corresponding to a continuation step. Again, the multilevel convergence is linear. The convergence of the linear system corresponding to the transport equation at this low Reynolds number is not slower than that for the stream function equation.

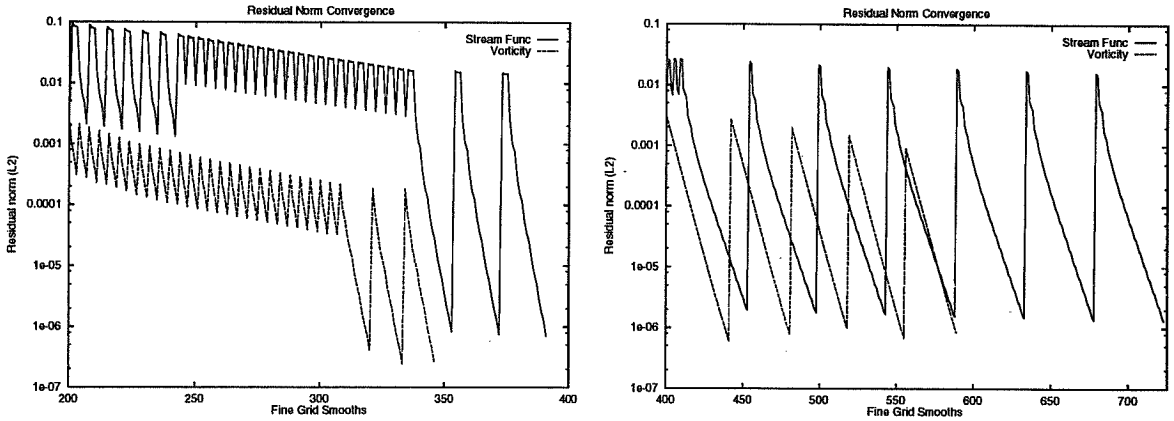


Figure 4: Multigrid convergence, stream function - vorticity, quadratic and quintic elements

Experiments were performed to test the performance of the nested iteration strategy on both model problems. The results indicate that the number of fine grid iterations is significantly reduced. The convergence behavior once the finest grid is reached is very similar to that shown previously for V-cycles without nested iteration.

Figure 7 shows the speedup on the Intel iPSC/860 hypercube for the stream function-vorticity problem. The speedups are presented for a grid of 1024 quadratic elements and a grid of 64 quintic elements, with a Lagrange basis used in both instances. The processor decomposition is performed by ordering the elements in the square domain naturally and distributing them to the processors in order. *i.e.* the first $\frac{N_e}{N_p}$ elements go to the first processor and so on, with N_e the number of elements and N_p the number of processors. The speedup for less than 16 processors is very good, with a parallel efficiency of .83 for 8 processors. The deterioration of performance above this level is due to the smaller problem sizes on each processor, meaning the communication-computation ratio is larger. The speedups are similar since the $p = 5$ case has fewer grid points (smaller problem size). For the same number of elements as for the $p = 2$ case, the speedup will obviously be better.

Conclusions

The focus of the present study has been the use of a multilevel scheme for preconditioning p-type finite element systems. We show that the spectral multilevel scheme serves as a useful preconditioner for the fine grid discretization resulting from the application of spectral finite elements. Convergence rates are linear for both chosen applications, and on both the self-adjoint and transport equations. A simple point Jacobi smoother can be used provided the correct relaxation is calculated for the corresponding problem and element degree. However, the study of more advanced smoothers, especially for the convection-dominated transport equations, is considered warranted.

Acknowledgments: This research has been supported in part by the Texas Advanced Technology Program and by the National Science Foundation.

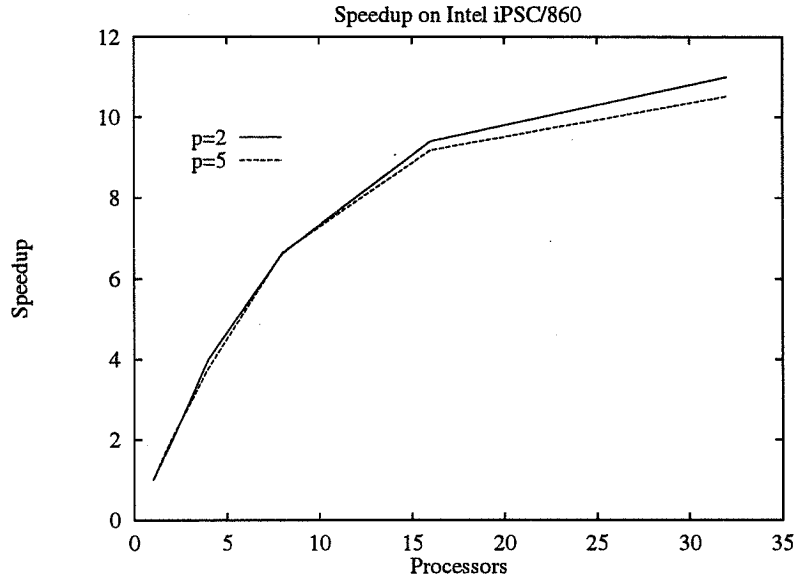


Figure 5: Speedup, stream function-vorticity, $Re = 100$, Lagrange basis

REFERENCES

[1] E. Barragy and G.F. Carey. A parallel element-by-element solution scheme. *Int. Jour. Num. Meth. Eng.*, 26:2367–2382, 1988.

[2] E. Barragy and G.F. Carey. Parallel-vector computations with high-p element-by-element methods. *Int. Jour. Comp. Math.*, 44:329–339, 1992.

[3] P.A. Blakey, X.L. Wang, C.M. Maziar, and P.A. Sandborn. A new technique for including velocity overshoot phenomena in conventional drift-diffusion simulators. In *Computational Electronics: Semiconductor Transport and Device Simulation*, pages 51–54, 1991.

[4] G.F. Carey. *Computational Grids: Generation, Refinement and Solution Strategies.* Wiley, 1996. In preparation.

[5] G.F. Carey, E. Barragy, R. McLay, and M. Sharma. Element-by-element vector and parallel computations. *Comm. App. Num. Meth.*, 4:299–307, 1988.

[6] G.F. Carey and B. Jiang. Element-by-element linear and nonlinear solution schemes. *Comm. App. Num. Meth.*, 2:145–153, 1986.

[7] M.B. Davis. *Parallel Multilevel Algorithms Applied to Iteratively Decoupled Transport Problems.* PhD thesis, University of Texas at Austin, 1996. In preparation.

[8] M.B. Davis and G.F. Carey. Parallel element-by-element spectral multilevel techniques. *Houston J. of Math*, 1996. In press.

[9] S. Foresti, G. Brussino, S. Hassanzadeh, and V. Sonnad. Multilevel solution method for the p-version of finite elements. *Comp. Phys. Comm.*, 53:349–355, 1989.

[10] H.K. Gummel. A self-consistent iterative scheme for one-dimensional steady state transistor calculations. *IEEE Trans. Elec. Dev.*, ED11:455–465, 1964.

[11] W. Hackbusch. *Multi-grid Methods and Applications.* Springer-Verlag, 1985.

[12] M. Lundstrom. *Modular Series on Solid State Devices, Volume X: Fundamentals of Carrier Transport.* Addison-Wesley, 1990.

[13] R.L. Panton. *Incompressible Flow.* Wiley, 1984.

[14] P. Roache. Finite difference methods for the steady-state Navier-Stokes equations. In *Proceedings of the Third International Conference on Numerical Methods in Fluid Mechanics*, volume 1, pages 139–145. Springer-Verlag, 1973.

[15] E.M. Ronquist and A.T. Patera. Spectral element multigrid. I. formulation and numerical results. *J. Sci. Comp.*, 2(4):389–406, 1987.

[16] S. Selberherr. *Analysis and Simulation of Semiconductor Devices.* Springer-Verlag, Wien, NY, 1984.

[17] C.M. Snowden. *Semiconductor Device Modelling.* Peter Peregrinus Ltd., London, 1988.

[18] P. Wesseling. *An Introduction to Multigrid Methods.* J. Wiley, Chichester, 1992.

Page intentionally left blank

REVENGE OF THE SEMICOARSENING FREQUENCY DECOMPOSITION MULTIGRID METHOD *

J. E. Dendy, Jr.
Theoretical Division, Los Alamos National Laboratory
Los Alamos, New Mexico 87545.

SUMMARY

The frequency decomposition multigrid method was previously considered and modified so as to obtain robustness for problems with discontinuous coefficients while retaining robustness for problems with anisotropic coefficients. The application of this modified method to a problem arising in global ocean modeling was also considered. For this problem it was shown that the discretization employed gives rise to an operator for which point relaxation is not robust. In fact, alternating line relaxation is required for robustness, negating the main advantage of the frequency decomposition method: robustness for anisotropic operators using only point relaxation. In this paper a semicoarsening variant, which requires line relaxation in one direction only, is considered, and it is shown that this variant works well for the global ocean modeling problem.

1 INTRODUCTION

Let us consider multigrid with standard coarsening on a rectangular grid of points; that is, the coarse grid offspring of a grid $\{x_{i,j} : i = 1,\ldots,m;\ j = 1,\ldots,n\}$ is the grid $\{x_{2i-1,2j-1} : i = 1,\ldots,\lceil m/2\rceil;\ j = 1,\ldots,\lceil n/2\rceil\}$. If point Gauss-Seidel with lexicographic ordering is the smoothing scheme, it is well-known [1] that degradation in convergence occurs for the usual five point discretization of

$$-aU_{xx} - bU_{yy} = F,$$

when $0 < a \ll b$ or when $0 < b \ll a$. One cure is to use line Gauss-Seidel as a smoother [1]. Another is to use semicoarsening instead of standard coarsening [2, 3, 4, 5]. Still another is to employ algebraic multigrid [6, 7]. Of these three, only algebraic multigrid also handles the case of the skew Laplacian, i.e.,

$$-\triangle^{sk,h}U_{i,j} = \frac{1}{2h^2}(4U_{i,j} - U_{i-1,j-1} - U_{i-1,j+1} - U_{i+1,j-1} - U_{i+1,j+1}), \tag{1.1}$$

but at the expense of having to use unstructured grids. Another multigrid scheme which handles both anisotropic coefficients and the skew Laplacian, using only standard coarsening and point Gauss-Seidel as the smoother, is the multigrid method considered by Brandt and Ta'asan [8]. The idea of the method, as described by Ta'asan, is as follows: when relaxation is slowly converging,

* **This work was performed under the auspices of the U.S. Department of Energy under contract W-7405-ENG-36 and was supported by the Office of Scientific Computing of the Department of Energy under Contract No. KC-07-01-01.**

the finest grid error must have the form

$$V = V_0 + \sum_{j=1}^{n} e^{iS_j} V_j,$$

where the V_j are smooth, the e^{iS_j} are highly oscillatory, and $n < 2^d$ (d being the dimension of the problem). "This error cannot be approximated on a coarser grid, because it is too oscillatory. Since [the] V_j are smooth functions, they can be approximated on the next coarser grid. Therefore, $n + 1$ coarse visits are done, each time solving for another V_j." [8] In the case that the V_j's correspond to $(0,0)$, $(0,\pi)$, $(\pi,0)$, and (π,π), Ta'asan argued that on the jth coarse grid visit, the coarse grid equation should approximate the equation

$$L_j^H V_j^H = R_j^H,$$

where

$$L_j^H = I_h^H e^{-i\underline{S}_j \cdot \underline{x}/h} L^h e^{i\underline{S}_j \cdot \underline{x}/h} I_H^h$$

and

$$R_j^H = I_h^H e^{-i\underline{S}_j \cdot \underline{x}/h} R^h,$$

where R^h is the residual on the fine grid with spacing h and I_H^h is bilinear interpolation from the coarse grid with spacing $H(= 2h)$ to the fine grid. In this case $e^{i\underline{S}_j \cdot \underline{x}/h} I_H^h$ is just I_H^h with some judicious sign changes. For specific cases, Ta'asan demonstrated that this methodology could be simplified so that the coarse grid operators could be formed directly instead of variationally. However, in the special case of V_j's corresponding to $(0,0)$ and (π,π), [9] follows the methodology just described.

A variant of Brandt and Ta'asan's method is the frequency decomposition multigrid method, developed independently by Hackbusch [10]. To describe this method, let us assume doubly periodic boundary conditions and suppose that the finest grid is the collection of points Ω^M shown in Fig. 1. Subdivide Ω^M into the four sets $\{\Omega_{k,l}^{M-1}, k = 0,1,\ l = 0,1\}$ as shown. Define $I_{k,l} : \Omega_{k,l}^{M-1} \to \Omega^M$ by

$$I_{k,l} = \frac{1}{4} \begin{pmatrix} (-1)^{k+l} & 2(-1)^l & (-1)^{k+l} \\ 2(-1)^k & 4 & 2(-1)^k \\ (-1)^{k+l} & 2(-1)^l & (-1)^{k+l} \end{pmatrix}, \tag{1.2}$$

where periodicity is invoked near the boundaries. Define $L_{k,l}^{M-1} = I_{k,l}^* L^M I_{k,l}$, and let $I_{k,l}^*$ be the residual operator, $I_{k,l}^* : \Omega^M \to \Omega_{k,l}^{M-1}$. Thus a two level method is given by:
1. Perform ν_1 multi-color Gauss-Seidel iterations on $L^M u^M = F^M$.
2. Solve $L_{k,l}^{M-1} V^{M-1} = f_{k,l}^{M-1} \equiv I_{k,l}^*(F^M - L^M u^M)$, $k = 0,1,\ l = 0,1$ directly.
3. Perform $u^M \leftarrow u^M + I_{k,l} V^{M-1}$, $k = 0,1,\ l = 0,1$.
4. Perform ν_2 multi-color Gauss-Seidel iterations on $L^M u^M = F^M$.

The frequency decomposition multigrid method is given by applying this process recursively. That is, instead of step 2, one decomposes each of the $\Omega_{k,l}^{m-1}$'s into four subsets and treats each of these with the two level process, continuing until the grids have few enough points that direct solution or solution by iteration alone is efficient.

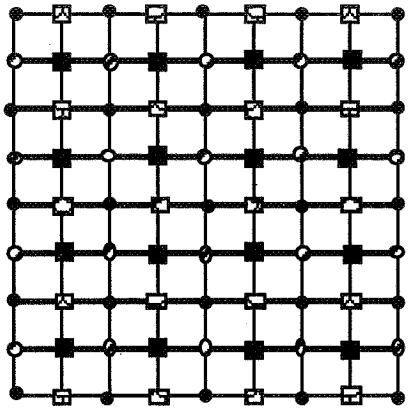


Fig. 1: $\Omega_{i,j}^{M-1}, i = 0,1,\ j = 0,1.$ $\Omega_{0,0}^{M-1}$ is designated by solid dots, $\Omega_{1,0}^{M-1}$ is designated by nonsolid squares, etc.

The frequency decomposition method is not robust for problems with discontinuous coefficients. In [11] we showed how to modify it to be robust for such problems while retaining robustness for problems with anisotropic coefficients. We also considered application of this modified method to a problem arising in global ocean modeling. For this problem it was shown that the discretization employed gives rise to an operator for which point relaxation is not robust. In fact, alternating line relaxation is required for robustness, negating the main advantage of the frequency decomposition method: robustness for anisotropic operators using only point relaxation. Given the necessity of performing alternating line relaxation, it is natural to consider a semicoarsening variant of the frequency decomposition multigrid method. In this variant, discussed in Section 2, the finest grid is coarsened only in the y-direction, and line relaxation by lines in x is performed. This variant is robust for constant coefficient, anisotropic problems, but it must be modified, as in [11], to be robust for problems with discontinuous coefficients. In Section 3 we consider the same numerical examples that were considered in [11]. In Section 4 we consider the application of this modified method to the same problem considered in [11] arising in global ocean modeling.

2 A SEMICOARSENING FREQUENCY DECOMPOSITION MULTIGRID METHOD

Let us consider multigrid with semicoarsening on a rectangular grid of points; that is, the coarse grid offspring of a grid $\{x_{i,j} : i = 1,\ldots,m;\ j = 1,\ldots,n\}$ is the grid $\{x_{i,2j-1} : i = 1,\ldots,m;\ j = 1,\ldots,\lceil n/2\rceil\}$. The robustness of line relaxation coupled with semicoarsening for constant coefficient anisotropic problems was first reported in [2]. For problems with anisotropic and discontinuous coefficients, a semicoarsening method was considered in [3] for three-dimensional problems. The two-dimensional analogue of this method is considered in [4] and [5]. Both of these papers use a technique due to Schaffer [12]; without this technique, the semicoarsening method would not be competitive. However, this method is not robust for operators like $-\triangle^{sk,h}$ in (1.1). (See the discussion in Section 1.)

To describe the semicoarsening frequency decomposition multigrid method (SFDM), let us

assume doubly periodic boundary conditions and suppose that the finest grid is the collection of points Ω^M shown in Fig. 1. Subdivide Ω^M into the two sets $\{\Omega_k^{M-1}, k = 0, 1\}$, where Ω_0^{M-1} is the set of odd x-lines and Ω_1^{M-1} the set of even lines of Ω^M. Define $I_k : \Omega_k^{M-1} \to \Omega^M$ by

$$I_k = \frac{1}{2} \begin{pmatrix} (-1)^k \\ 2 \\ (-1)^k \end{pmatrix}, \tag{2.1}$$

where periodicity is invoked near the boundaries. Define $L_k^{M-1} = I_k^* L^M I_k$ and let I_k^* be the residual weighting operator, $I_k^* : \Omega^M \to \Omega_k^{M-1}$. Thus a two level method is given by:
1. Perform ν_1 red-black Gauss-Seidel line iterations, by lines in x, on $L^M u^M = F^M$.
2. Solve $L_k^{M-1} V^{M-1} = f_k^{M-1} \equiv I_k^*(F^M - L^M u^M)$, $k = 0, 1$ directly.
3. Perform $u^M \leftarrow u^M + I_k V^{M-1}$, $k = 0, 1$.
4. Perform ν_2 red-black Gauss-Seidel line iterations, by lines in x, on $L^M u^M = F^M$.

The semicoarsening frequency decomposition multigrid method is given by applying this process recursively. That is, instead of step 2, one decomposes each of the Ω_k^{M-1}'s into two subsets and treats each of these with the two level process, continuing until the coarsest grid consists of a collection of decoupled sets, each set consisting of just one x-line.

Since the frequency decomposition method is not robust for problems with discontinuous coefficients, one would hardly expect SFDM to be robust for such problems. We use the same numerical example employed in [11] to show in Section 3 that this expectation is justified. The key ingredient for obtaining robustness for problems with discontinuous coefficients is to use operator induced interpolation. The other ingredient is to use Galerkin coarsening, but that ingredient is already present here.

Let us first recall how operator-induced interpolation is defined in the case of semicoarsening [4, 12, 5] for nine point operators. It suffices to consider the two level method; let the template of L^M at a given point be

$$\begin{pmatrix} NW & N & NE \\ W & C & E \\ SW & S & SE \end{pmatrix}. \tag{2.2}$$

For this discussion we do not need to introduce indices. For step 3 above, I_0 is just the identity for odd lines; for even lines, let

$$A^- V^- + A^0 V^0 + A^+ V^+ = 0$$

be the equation that would give the row $V^0 = (V_{i,j} : i = 1, \cdots, M)$ in terms of the rows $V^- = (V_{i,j-1} : i = 1, \cdots, M)$ and $V^+ = (V_{i,j+1} : i = 1, \cdots, M)$, for j even. Here A^-, A^0, and A^+ are all tridiagonal matrices;

$$\begin{aligned} A^- &= tridiag(SW\ S\ SE), \\ A^0 &= tridiag(W\ C\ E), \\ \text{and } A^+ &= tridiag(NW\ N\ NE). \end{aligned} \tag{2.3}$$

Then

$$V^0 = -(A^0)^{-1}(A^- V^- + A^+ V^+). \tag{2.4}$$

Unfortunately, use of (2.4) yields a nonsparse interpolation, leading to nonsparse coarse grid operators. Schaffer's idea [12] is to assume that $-(A^0)^{-1}A^-$ and $(-A^0)^{-1}A^+$ can each be approximated by diagonal matrices in the sense that B^- and B^+ are diagonal matrices such that

$$-(A^0)^{-1}A^- e = B^- e \text{ and } (-A^0)^{-1}A^+ e = B^+ e,$$

where e is the vector $(1,\cdots,1)^T$. To find B^- and B^+ requires just two tridiagonal solves. The interpolation formula for I_0 using B^- and B^+ is

$$V^0 = B^- V^- + B^+ V^+.$$

The derivation for I_1 is a bit different. If we consider just Ω_0^{M-1}, we have an ordinary semicoarsening multigrid method, which is robust for operators which annihilate $(0,0)$, $(0,\pi)$, and $(\pi,0)$. It is not robust for operators which annihilate (π,π); to obtain such robustness is the role of Ω_1^{M-1}. We can repeat the above argument, except that now e is the vector $(-1,1,-1,1,\cdots)^T$. The interpolation formula for I_1 using the resulting B^- and B^+ is

$$V^0 = -|B^-|V^- - |B^+|V^+, \tag{2.5}$$

where $|B^+|$ $[|B^-|]$ is the diagonal matrix whose entries are the absolute values of the corresponding entries of $B^+[B^-]$.

It can be checked that in the case of constant coefficient zero-sum nine point difference operators, this construction gives (2.1). The same procedure is used recursively in the multigrid case. We use the notation

$$\Omega^{M-k}_{j_1,\cdots,j_{2^k}}, j_i = 0 \text{ or } 1 \tag{2.6}$$

to denote the general level $M-k$ grid, $k = 1,\ldots,M-1$. In analogy with the terminology used in [11] we refer to this modification of SFDM as CSFDM for "child of the semicoarsening frequency decomposition multigrid method."

There are some problems for which the presence of Ω_1^{M-1} contaminates the solution process and leads to slower convergence. Examples are given in Section 3. An analogous situation occurs in [11]. There the solution was to design switches to detect the strength of certain frequencies and to include the corresponding corrections with strength ϕ, $0 \le \phi \le 1$. The same solution is employed here. Consider Ω_1^{M-1}. Define

$$\phi = max(0, 1 - \frac{|C+SW+NW+SE+NE|}{|C+W+S+E+N|}).$$

(In this description we ignore the possibility of zero divides to simplify the exposition.) Thus, (2.5) is replaced by

$$V^0 = \phi(-|B^-|V^- - |B^+|V^+).$$

Note that ϕ is 0 for the standard five point discretization of the Laplacian and 1 for $-\triangle^{sk,h}$. We refer to this modification of CSFDM as GSFDM, for grandchild of the semicoarsening frequency decomposition multigrid method.

3 NUMERICAL EXAMPLES

All of these examples appeared in [11]. They are for problems that are 64 x 64 in size, this size problem being sufficient to illustrate the points we are making. We consider five problems. The first is

$$-\nabla \cdot (D(x,y)\nabla U(x,y)) + \sigma(x,y)U(x,y) = F(x,y)$$

in a bounded region Ω of R^2, where $D = (D^1, D^2)$, D^i is positive, $i = 1, 2$, and D^i, σ, and F are allowed to be discontinuous across internal boundaries Γ of Ω; moreover, $D_1 \gg D_2$ and $D_1 \ll D_2$ in different subregions of Ω is possible. Specifically, we consider

$$\begin{cases} -\nabla \cdot (D\nabla U) + U = F \text{ on } (0, 16.) \text{ x } (0., 16.) \\ U \text{ doubly periodic,} \end{cases} \tag{3.1}$$

for the region shown in Fig. 2 and for the values of $D = D^1 = D^2$ and F indicated there. The differencing employed is given in [13].

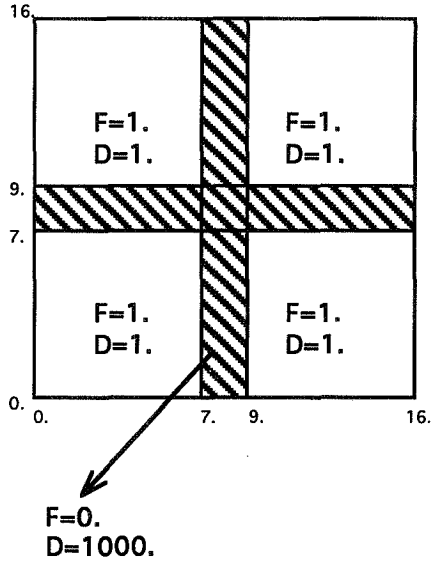


Fig. 2: Diffusion coefficients and right hand side for (3.1)

The second is the standard discretization of

$$\begin{cases} -U_{xx} - .00001 U_{yy} = F \text{ on } (0, 16.) \text{ x } (0., 16.) \\ U \text{ doubly periodic,} \end{cases} \tag{3.2}$$

where F is chosen so that $\int F = 0$, specifically

$$F(x, y) = \begin{cases} 1. \text{ if } 0. \le y \le 4. \text{ or } 12. \le y \le 16. \\ -1. \text{ otherwise.} \end{cases} \tag{3.3}$$

The third problem is

$$\begin{cases} -\triangle^{sk,h} U = F \text{ on } (0, 16.) \text{ x } (0., 16.) \\ U \text{ doubly periodic,} \end{cases} \tag{3.4}$$

where $\triangle^{sk,h}$ is given in (1.1) and F is given in (3.3). We note that for (3.4) to have a solution, F must also satisfy $\sum_{i,j} (-1)^i (-1)^j F_{i,j} = 0$; this condition is fortuitously satisfied by (3.3).

The fourth problem has anisotropic and discontinuous coefficients,

$$\begin{cases} -\nabla \cdot (D\nabla U) + U = F \text{ on } (0., 16.) \text{ x } (0., 16.) \\ U \text{ doubly periodic,} \end{cases} \tag{3.5}$$

with the coefficients and right hand side given in Fig. 3. The differencing employed is given in [13].

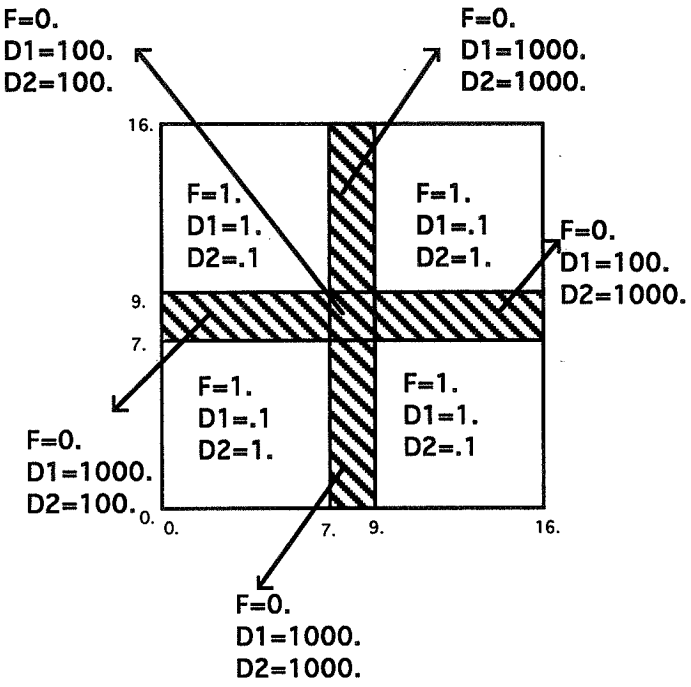


Fig. 3: Diffusion coefficients and right hand side for (3.4)

The fifth problem comes from [8]. We consider the operator $L^{h,\alpha}$ with template

$$\frac{1}{h^2}\begin{pmatrix} -\frac{1}{2} & -\alpha & -\frac{1}{2} \\ \alpha & 2 & \alpha \\ -\frac{1}{2} & -\alpha & -\frac{1}{2} \end{pmatrix},$$

where $|\alpha| < 1$. We consider

$$\begin{cases} L^{h,.95}U = F \text{ on } (0,16.) \text{ x } (0.,16.) \\ U \text{ doubly periodic,} \end{cases} \tag{3.6}$$

where F is given in (3.3).

Table 1 shows the results for SFDM for these five problems. The first column indicates the problem, the second the number of V-cycles (less than eleven) to solve until the final residual r satisfies $\|r\| \leq 10^{-6}$, the third the convergence factor of the first cycle, the fourth the convergence factor of the last cycle, and the last the average convergence factor. (Recall that the average convergence factor for p V-cycles is defined as $(\|r_p\|/\|r_0\|)^{\frac{1}{p}}$, where $\|\cdot\|$ is the discrete L^2 norm, and r_k is the residual on the finest grid after k V-cycles.) An initial guess of zero is used. Red-black line relaxation by lines in x is used on all grids. The V-cycle employed uses $\nu_1 = \nu_2 = 1$. In Tables 2 and 3 we give the same data for CSFDM and GSFDM. One can see that CSFDM and GSFDM perform much better than SFDM for (3.1) without degradation in convergence factor for the other problems. The difference in CFDM and GFDM is dramatic only for (3.2) in contrast

to the corresponding methods in [11]. Convergence factors for the semicoarsening variants are comparable to those in [11] and even significantly better for (3.2) and (3.5).

TABLE 1:PERFORMANCE OF SFDM FOR FIVE PROBLEMS

Problem	Number of Cycles	CF — First Cycle	CF — Last Cycle	average CF
(3.1)	10*	.38	.55	.53
(3.2)	7	.08	.08	.08
(3.4)	7	.08	.08	.08
(3.5)	10*	.34	.59	.56
(3.6)	6	.04	.07	.05

* fails to converge in ten cycles

TABLE 2:PERFORMANCE OF CSFDM FOR FIVE PROBLEMS

Problem	Number of Cycles	CF — First Cycle	CF — Last Cycle	average CF
(3.1)	9	.10	.15	.14
(3.2)	7	.08	.08	.08
(3.4)	7	.08	.08	.08
(3.5)	10*	.10	.21	.19
(3.6)	6	.04	.07	.05

* fails to converge in ten cycles

TABLE 3:PERFORMANCE OF GSFDM FOR FIVE PROBLEMS

Problem	Number of Cycles	CF — First Cycle	CF — Last Cycle	average CF
(3.1)	9	.08	.14	.13
(3.2)	1	3.0×10^{-9}	3.0×10^{-9}	3.0×10^{-9}
(3.4)	6	.03	.06	.05
(3.5)	10	.08	.19	.18
(3.6)	6	.03	.06	.05

The coarsest grid problem in all three variants consists of a collection of decoupled sets, each set consisting of just one x-line. If the problem is nonsingular, there is no difficulty in solving the associated periodic tridiagonal systems. If the problem is singular, then the tridiagonal system for $\Omega^1_{0,\ldots,0}$ (see (2.6)) is singular. To attain uniqueness one need only add a positive number to one of the diagonals of this tridiagonal system, pinning down the solution for this grid and thus assuring a unique solution. Such a problem, of course, has a solution determined only up to a constant; i.e., the computed solution plus any constant is still a solution. A similar technique is used in [14]. In the case of (1.1), the tridiagonal system for $\Omega^1_{1,\ldots,1}$ is singular; addition of a positive number to one of the diagonals is all that is required in this case as well.

For parallelization of SFDM and its offspring on the CM-5, we lay the grids out in the obvious way. Efficient communication in x on all grids is obvious since each point communicates only with its nearest left and right neighbors. Communication in y is efficient since each point communicates with bottom and top neighbors a power of two distant, and with the immediate left and right neighbors of these points. Like the methods in [11, 15, 10], the methods here keep all the processors busy on every grid level, and again this busyness is actually a disadvantage when the number of points per processor exceeds one (vp ratio greater than one), for then the virtual processors are kept busy on every level as well. In the method of [4], work is halved on each coarser level until a vp ratio of one is reached: from then on, work on each level remains constant, with more and more processors becoming idle. But for SFDM and its offspring, work on each level remains constant regardless of the vp ratio. For the method of [4] and a vp-ratio > 1, it is possible to organize the problem so that efficient relaxation can be achieved per processor and — by doing intra-processor moves — still achieve efficiency for interpolation and residual weighting; most of the communication is done within individual processors, not between processors. But for SFDM and its offspring as organized here, for sufficiently coarse levels, one is forced to pay the same off processor communication penalty for every point of every grid.

4 APPLICATION TO A GLOBAL OCEAN MODELING PROBLEM

The original motivation for this work came from an application in global ocean modeling. In [16] an elliptic equation is solved at each time step. This equation is differenced so that the (π, π) frequency is in the null space of the operator. The reason for this differencing is that it is required for an energy conservation relation that is deemed to be important to long time integration of the system. This differencing is common in the meteorological community, although some rebels are attempting to introduce new models which do not employ it. There are other difficulties as well. Since spherical coordinates are employed (fortunately with the regions near the poles left out), the difference stencil (when normalized) is close to $L^{h,\alpha}$ (see (3.5)), with $|\alpha|$ close to 1, in some regions. The diffusion coefficient depends on the depth of the ocean. On the scale of the grids used, this depth jumps no more than a factor of a hundred from cell to cell. Land masses are dealt with by the use of dead cells; that is, on land the equation that is solved is $(Id)U = 0.$, where Id is the identity operator. The presence of dead cells and discontinuous coefficients really rules out the use of SFDM. Both CSFDM and GSFDM provide a mechanism for assuring that the coarse grid dead cells do not couple to the coarse grid ocean cells. The final difficulty is that the boundaries, approximated by lines of constant latitude and longitude, are ragged — coastlines tend to be fractal.

Because of the existence of lines of latitude that intersect no land masses, for which periodic boundary conditions are imposed, we need an efficient solver for periodic tridiagonal systems. Such a solver is still not available in CMSSL (Connection Machine Scientific Software Library). Thus we still employ a trick due to R. D. Richtmyer [17]: Let the unknowns of the periodic tridiagonal system be indicated by $\{x_1, \cdots, x_m\}$. Set $x_m = 0.$, and solve for $\{x_1, \cdots, x_{m-1}\}$, denoting the solution by s^0. Set $x_m = 1.$, and solve for $\{x_1, \cdots, x_{m-1}\}$, denoting the solution by s^1. (The CMSSL tridiagonal solution algorithms can be used to solve for s^0 and s^1.) Every linear combination of s^0 and s^1 has zero residual for $\{2, \cdots, m-1\}$. It is easy to construct the linear combination that has zero residual at 1 and m as well. This linear combination involves division by the difference of residuals of the system at 1 for s^0 and s^1; this can involve the difference of two small, nearly

equal numbers and lead to the tridiagonal system being solved to not very great precision. The cure is to use the obvious defect correction algorithm to obtain more digits of accuracy. In [11] the better conditioning of the coarse grid operators (in comparison with the operators obtained from semicoarsening) resulted in not having to use this defect correction algorithm.

In the original model, the solution of a steady state, zero row-sum, discrete, elliptic equation, call it $L^h U^h = F^h$, was required at each time step. The problem of generating a compatible right hand F^h for testing was solved by applying the difference operator to a random grid function; the F^h thus generated satisfies $\sum_{i,j} F^h_{i,j} = 0$ and $\sum_{i,j}(-1)^i(-1)^j F^h_{i,j} = 0$. In [11] many simplified situations were investigated, with the intent of showing that the reason for poor convergence for the actual problem was poor approximation on coarse grids due to the complicated boundary. We omit the investigation of these simplified situations here since the behavior of the semicoarsening variants parallels the behavior of the methods in [11].

The original model was improved by requiring the solution of a time-dependent equation [18]. Thus at the nth time step, one must solve

$$\frac{G}{(\Delta t)^2} U^{h,n} + L^h U^{h,n} = F^{h,n}, \tag{4.1}$$

where $G_{i,j}$ = const.(area of (i,j)th cell). In this model the size of the time step, Δt, is limited by a Courant condition. For the 256 x 128 problem considered here, the ratio of $\frac{G}{(\Delta t)^2}$ to the diagonal of L^h ranges from .01 to 35.0 for the active cells, with a mean value, including dead cells, of .3. There is no apparent correlation of the value of this ratio with the location of the boundaries, but it was clear in [11] that the addition of this time step term to the operator greatly improves the correction capabilities of the coarse grid operators. However, it was also shown that the time step is not large enough to achieve a good convergence factor with relaxation alone. As in Section 3, a zero initial guess is used.

TABLE 4:PERFORMANCE FOR THE GLOBAL OCEAN PROBLEM (4.1)

Problem	Number of Cycles	CF — First Cycle	CF — Last Cycle	average CF
CSFDM	10*	.03	.60	.34
GSFDM	10*	.03	.63	.35
CSFDMA	10	.02	.42	.27
CSFDMB	5	.02	.12	.05
CSFDMC	5	.02	.11	.05

* fails to converge in ten cycles

The performance of CSFDM and GSFDM is in sharp contrast to the situation in [11], where the addition of the time step term results in great convergence. There are three variants of CSFDM listed in Table 4, CSFDMA, CSFDMB, and CSFDMC, the last two of which give convergence equal to what was achieved in [11] with alternating line relaxation. To motivate and explain these variants, it is necessary to recall the construction of operator induced interpolation in the case of standard coarsening black box multigrid [13, 9, 19, 14]: At coarse grid points coinciding with fine grid points, interpolation is just the identity. At a fine grid point lying vertically between

two coarse grid points, interpolation at $v_{i,j}$ is given by $av_{i,j-1} + bv_{i,j+1}$, where

$$a = -(SW + S + SE)/(W + C + E) \text{ and } b = -(NW + N + NE)/(W + C + E) \tag{4.2}$$

and where we have used the notation of (2.2). That is, one thinks of summing away the x-dependence to obtain a three point relation between $v_{i,j-1}$, $v_{i,j}$, and $v_{i,j+1}$. A difficulty with this approach, when using standard coarsening, is that if $\rho = C + NW + N + NE + W + E + SW + S + SE$ is small, then instead of using $W + C + E$ in (4.2), one should use $-SW - S - SE - NW - N - NE$ instead; this point is discussed in [14].

In semicoarsening black box multigrid, the analogous choice would be to use

$$-NW - N - NE - SW - S - SE - W - E \tag{4.3}$$

instead of C in (2.3). In normal semicoarsening black box multigrid [4] (and for Ω_0^{M-1} here), however, this choice leads to no improvement in convergence factor. For Ω_1^{M-1}, the analogous choice is to use

$$|-NW - W - SW + S + N - SE - E - NE| \tag{4.4}$$

instead of C in (2.3) to derive (2.5); this choice on coarser grids can lead to operators for which $C + NW + N + NE + W + E + SW + S + SE > 0$ is no longer valid. Hence, it seems safer to use (4.4) only for level M.

To summarize, in Table 4 CSFDMA, CSFDMB, and CSFDMC have the following meaning:
CSFDMA: CSFDM with (4.3) and (4.4) enforced at all levels.
CSFDMB: CSFDM with (4.4) enforced at all levels.
CSFDMC: CSFDM with (4.4) enforced only at level M.

Simple analysis shows what can go wrong with using C instead of (4.4). Let us consider the operator

$$L = \begin{pmatrix} -\frac{1}{2} & 1-\epsilon & -\frac{1}{2} \\ -1+\epsilon & 2+\eta & -1+\epsilon \\ -\frac{1}{2} & 1-\epsilon & -\frac{1}{2} \end{pmatrix},$$

where η and ϵ are both nonnegative and small. In this case, if we use C instead of (4.3), $|B^-|$ and $|B^+|$ in (2.5) are both $diag(\theta)$, where $\theta = \frac{1}{2+\frac{\eta}{\epsilon}}$. Thus $0 \le \theta \le \frac{1}{2}$. A computation shows that $I_1^* L I_1$ has the form (2.2), where $C = (1-2\theta+4\theta^2)+(1+2\theta^2)\eta+(1+2\theta)\epsilon$, $W = E = -\frac{1}{2}+\frac{3}{2}\epsilon$, $S = N = -2\theta+2\theta^2+\theta^2\eta+2\theta\epsilon$, and $SW = NW = NE = SE = \theta - \theta^2 + \theta^2\epsilon$. For η sufficiently large, $W + C + E$ is positive and A^0 in (2.3) is invertible. For $\epsilon = .05$ and $\eta = .02$, $\theta = \frac{5}{12}$, and $C + W + E$ is negative. Thus by continuity, $C + W + E$ is zero for some values of η and ϵ, and A^0 is singular; for nearby values of η and ϵ, A^0 is nearly singular, and ill-conditioning occurs. If we use (4.3), however, then $\theta = \frac{1}{2}$, $E = W = -\frac{1}{2} + \frac{3}{2}\epsilon$, $C = 1 + \frac{3}{2}\eta + \epsilon$, and $W + C + E = \frac{3}{2}\eta + 5\epsilon$ is always positive. Similar arguments show that if (4.3) is always used, then $C+NW+N+NE+W+E+SW+S+SE < 0$ can happen on coarser grids. Numerical experiments show that this seems to happen on grids of the form $\Omega_{j_1,\ldots,j_k}^{M-k}$ (see (2.6)), where $k \ge 3$ and $j_i = j_{i+1} = 1$ for some i. Such grids can be deleted [20, 8] without harming the convergence factor, as illustrated by the nearly identical performance of CSFDMB and CSFDMC. Thus an alternative would be to include the corrections from such grids with weight zero.

REFERENCES

[1] A. Brandt, *Multi-level adaptive solutions to boundary-value problems*, Math. Comp. 31(1977), pp. 333-390.

[2] G. Winter, *Fourienanalyse zur Konstruktion schneller MGR-Verfahren*, Ph.D. Thesis, Rheinischen Friedrich-Wilhelms-Universität zu Bonn, 1982.

[3] J. E. Dendy, Jr., S. F. McCormick, J. W. Ruge, T. F. Russell, S. Schaffer, *Multigrid methods for three-dimensional petroleum reservoir simulation*, Proceedings of the Tenth Symposium on Reservoir Simulation, Houston, TX, Feb. 6-8, 1989, pp. 19-25.

[4] J. E. Dendy, Jr., M. P. Ida, and J. M. Rutledge, *A semicoarsening multigrid algorithm for SIMD machines*, SIAM J. Sci. Stat. Comp., 13[1992], pp. 1460-1469.

[5] R. A. Smith, A. Weiser, *Semicoarsening multigrid on a hypercube*, SIAM J. Sci. Stat. Comp., 13(1992), pp. 1314-1329.

[6] A. Brandt, S. McCormick, J. Ruge, *Algebraic multigrid (AMG) for automatic multigrid solution with application to geodetic computations*, manuscript, 1983.

[7] K. Stüben, *Algebraic multigrid: experiences and comparisons*, Appl. Math. Comp. 13(1983), pp. 419-451.

[8] S. Ta'asan, *Multigrid methods for highly oscillatory problems*, Ph.D. Thesis, Weizmann Institute of Science, 1984.

[9] J. E. Dendy, Jr., *Multigrid semi-implicit hydrodynamics revisited*, in Large Scale Scientific Computation, Seymour V. Parter, editor, pp. 1-22, Academic Press, New York, 1984.

[10] W. Hackbusch, *The frequency decomposition multigrid method, part I: application to anisotropic equations*, Numer. Math. 56(1989), pp. 229-245.

[11] J. E. Dendy, Jr. and C. T. Tazartes, *Grandchild of the frequency decomposition multigrid method*, to appear in SIAM J. Sci. Comput.

[12] S. Schaffer, *A semicoarsening multigrid method for elliptic partial differential equations with highly discontinuous and anisotropic coefficients*, to appear.

[13] R. E. Alcouffe, A. Brandt J. E. Dendy, Jr., and J. W. Painter, *The multi-grid method for the diffusion equation with strongly discontinuous coefficients*, SIAM J. Sci. Stat. Comp. 2(1981), pp. 430-454.

[14] J. E. Dendy, Jr. *Black box multigrid for periodic and singular problems*, Appl. Math. Comp. 25(1988), pp. 1-10.

[15] P.O. Frederickson and O.A. McBryan, *Normalized convergence rates for the PSMG method*, SIAM J. Sci. Statist. Comput, 12(1981), pp. 221-229.

[16] R. D. Smith, J. K. Dukowicz and R. C. Malone, *Parallel ocean general circulation modeling*, Physica D, 60(1992), pp. 38-61.

[17] R. D. Richtmyer, Talk given at Los Alamos National Laboratory in the deadest past of the 1970's.

[18] J. K. Dukowicz and R. D. Smith, *Implicit free-surface method for the Bryan-Cox-Semtner Ocean Model*, J. Geophys. Res. 99(1994), pp. 7991-8014.

[19] J. E. Dendy, Jr., *Black box multigrid*, J. Comp. Phys. 48(1982), pp. 366-386.

[20] W. Hackbusch, *The frequency decomposition multigrid method*, in Robust Multigrid Methods, Proceedings of the Fourth GAMM Seminar, Notes on Numerical Fluid Mechanics, Vol. 23, Vieweg Verlag, Braunschweig, FRG, 1988.

Page intentionally left blank

AN OPTIMAL ORDER NONNESTED MIXED MULTIGRID METHOD FOR GENERALIZED STOKES PROBLEMS

QINGPING DENG†

Abstract. A multigrid algorithm is developed and analyzed for generalized Stokes problems discretized by various nonnested mixed finite elements within a unified framework. It is abstractly proved by an element-independent analysis that the multigrid algorithm converges with an optimal order if there exists a "good" prolongation operator. A technique to construct a "good" prolongation operator for nonnested multilevel finite element spaces is proposed. Its basic idea is to introduce a sequence of auxiliary nested multilevel finite element spaces and define a prolongation operator as a composite operator of two single grid level operators. This makes not only the construction of a prolongation operator much easier (the final explicit forms of such prolongation operators are fairly simple), but the verification of the approximate properties for prolongation operators is also simplified. Finally, as an application, the framework and technique is applied to seven typical nonnested mixed finite elements.

Key words. generalized Stokes problems, mixed methods, multigrid algorithm, nonnested

AMS(MOS) subject classifications. 65F10, 65N30

1. Introduction. This paper will develop an optimal order multigrid algorithm for solving mixed finite element equations of the following generalized Stokes problems:

$$\begin{cases} -\Delta \underset{\sim}{u} + \nabla p = \underset{\sim}{f}, & \text{in } \Omega, \\ \operatorname{div} \underset{\sim}{u} = g, & \text{in } \Omega, \\ \underset{\sim}{u} = 0, & \text{on } \partial\Omega. \end{cases} \tag{1.1}$$

where Ω is a bounded convex domain in $\mathbb{R}^2$. If $\underset{\sim}{f} \in H^{-1}(\Omega)$ and $g \in L_0^2(\Omega)$, (1.1) is uniquely solvable (cf. [20]). We refer to [7] and [20] for notations and definitions of the function spaces used in this paper. The velocity–pressure variational formulation of the saddle problem for (1.1) is to find $[\underset{\sim}{u}, p] \in (H_0^1(\Omega))^2 \times L_0^2(\Omega)$ such that

$$\mathcal{L}([\underset{\sim}{u}, p], [\underset{\sim}{v}, q]) = F([\underset{\sim}{v}, q]), \quad \forall\, [\underset{\sim}{v}, q] \in (H_0^1(\Omega))^2 \times L_0^2(\Omega), \tag{1.2}$$

where $(\cdot,\cdot) \equiv (\cdot,\cdot)_\Omega$ stands for the inner product in $L^2(\Omega)$ or $(L^2(\Omega))^2$, and

$$\mathcal{L}([\underset{\sim}{u}, p], [\underset{\sim}{v}, q]) = (\nabla \underset{\sim}{u}, \nabla \underset{\sim}{v}) - (p, \operatorname{div} \underset{\sim}{v}) - (q, \operatorname{div} \underset{\sim}{u}), \tag{1.3}$$

$$F([\underset{\sim}{v}, q]) = (\underset{\sim}{f}, \underset{\sim}{v}) + (g, q). \tag{1.4}$$

Let $\mathcal{T}_k$ $(k \geq 0)$ be a quasi-uniform triangular or rectangular partition of Ω with mesh size $h_k = h_0 2^{-k}$. $\mathcal{T}_k$ is obtained by linking the midpoints of the three edges of all triangles

†Department of Mathematics, The University of Tennessee, Knoxville, TN 37996. deng@math.utk.edu

of $\mathcal{T}_{k-1}$ or by linking the midpoints of two opposite sides of all rectangles of $\mathcal{T}_{k-1}$. For simplicity, we also assume that $\overline{\Omega} = \cup_{K\in\mathcal{T}_k}\overline{K}$. Let $\underset{\sim}{X}_k \subset (L^2(\Omega))^2$, $M_k \subset L_0^2(\Omega)$ be two finite element approximate spaces of $(H_0^1(\Omega))^2$ and $L_0^2(\Omega)$ associated with $\mathcal{T}_k$. The mixed finite element method for (1.2) at level k is to find $[\underset{\sim}{u}_k, p_k] \in \underset{\sim}{X}_k \times M_k$ such that

$$\mathcal{L}_k([\underset{\sim}{u}_k, p_k], [\underset{\sim}{v}, q]) = F_k([\underset{\sim}{v}, q]), \quad \forall\, [\underset{\sim}{v}, q] \in \underset{\sim}{X}_k \times M_k, \tag{1.5}$$

where $(\cdot,\cdot)_k = \sum_{K\in\mathcal{T}_k} (\cdot,\cdot)_K$, and

$$\mathcal{L}_k([\underset{\sim}{u}, p], [\underset{\sim}{v}, q]) = (\nabla \underset{\sim}{u}, \nabla \underset{\sim}{v})_k - (p, \operatorname{div} \underset{\sim}{v})_k - (q, \operatorname{div} \underset{\sim}{u})_k, \tag{1.6}$$

$$F_k([\underset{\sim}{v}, q]) = (\underset{\sim}{f}, \underset{\sim}{v})_k + (g, q)_k. \tag{1.7}$$

It is well-known that $\underset{\sim}{X}_k$ and M_k must satisfy the Babuška–Brezzi condition, i.e.,

$$\sup_{\underset{\sim}{v}_k \in \underset{\sim}{X}_k} \frac{|(q,\ \operatorname{div} \underset{\sim}{v}_k)_k|}{\|\underset{\sim}{v}_k\|_k} \geq \gamma_0 \|q\|_{L^2(\Omega)}, \quad \forall\, q \in M_k, \tag{1.8}$$

where $\|\underset{\sim}{v}_k\|_k^2 = (\nabla \underset{\sim}{v}_k, \nabla \underset{\sim}{v}_k)_k$, and γ_0 is a positive number independent of k and h_k. We also assume that the following error estimate and interpolation property hold (cf. [13],[20]):

$$\|\underset{\sim}{u} - \underset{\sim}{u}_k\|_{L^2(\Omega)} + h_k(\|\underset{\sim}{u} - \underset{\sim}{u}_k\|_k + \|p - p_k\|_{L^2(\Omega)}) \leq C h_k^2(\|\underset{\sim}{u}\|_{H^2(\Omega)} + \|p\|_{H^1(\Omega)}), \tag{1.9}$$

$$\begin{aligned}&\|\underset{\sim}{v} - \Pi_k \underset{\sim}{v}\|_{L^2(\Omega)} + h_k(\|\underset{\sim}{v} - \Pi_k \underset{\sim}{v}\|_k + \|q - \pi_k q\|_{L^2(\Omega)}) \\ &\quad \leq C h_k^2(\|\underset{\sim}{v}\|_{H^2(\Omega)} + \|q\|_{H^1(\Omega)}), \quad \forall\, [\underset{\sim}{v}, q] \in (H^2(\Omega) \cap H_0^1(\Omega))^2 \times (H^1(\Omega) \cap L_0^2(\Omega)).\end{aligned} \tag{1.10}$$

Here $\Gamma_k = [\Pi_k, \pi_k]$ is the interpolation operator associated with $\underset{\sim}{X}_k \times M_k$, $[\underset{\sim}{u}, p] \in (H^2(\Omega) \cap H_0^1(\Omega))^2 \times (H^1(\Omega) \cap L_0^2(\Omega))$ is the solution of (1.2), and $[\underset{\sim}{u}_k, p_k] \in \underset{\sim}{X}_k \times M_k$ is the solution of (1.5).

However, most commonly used low order mixed elements, which have a matched approximate order, do not satisfy (1.9). So, we have to modify them by using some special techniques, such as bubble functions, nonconforming elements, and composite elements. Unfortunately, the first two techniques must cause the nonnestedness of multilevel finite element spaces, and so does the third one for many cases. Hence, it is of interest and importance to study nonnested multigrid algorithms for mixed finite element equations of (1.1). Simultaneously, we have to overcome some new difficulties since the standard multigrid theory cannot be directly applied, and a prolongation operator other than the natural injection must be chosen. However, we observe that usually only the finite element velocity spaces are modified and the finite element pressure spaces are still some common finite element spaces. Therefore, the multilevel finite element pressure spaces are still

nested. In view of this observation, we always assume that the nestedness of multilevel finite element pressure spaces holds.

The objective of this paper is to develop and analyze an optimal order multigrid algorithm in a unified framework for finite element equation (1.5). The convergence of the multigrid algorithm is proved by an element-independent analysis. The technique to construct a "good" prolongation operator for nonnested multilevel spaces, that is, a prolongation operator which satisfies conditions (1.11) and (1.12), is proposed. The idea of defining the multigrid algorithm is adopted from [20]. Our convergence analysis mainly relies on the properties (1.11) and (1.12) of the prolongation operator $I_{k-1}^k : \underset{\sim}{X}_{k-1} \times M_{k-1} \to \underset{\sim}{X}_k \times M_k$ given as follows:

$$(1.11)\quad \|[\underset{\sim}{v}, q] - I_{k-1}^k[\underset{\sim}{v}, q]\|_{0,k} \le Ch_k(\|\underset{\sim}{v}\|_{k-1} + \|q\|_{L^2(\Omega)}), \quad \forall\, [\underset{\sim}{v}, q] \in \underset{\sim}{X}_{k-1} \times M_{k-1},$$

$$(1.12)\quad \|[\underset{\sim}{v}, q] - I_{k-1}^k \Gamma_{k-1}[\underset{\sim}{v}, q]\|_{0,k} \le Ch_k^2(\|\underset{\sim}{v}\|_{H^2(\Omega)} + \|q\|_{H^1(\Omega)}),$$
$$\forall\, [\underset{\sim}{v}, q] \in (H^2(\Omega) \cap H_0^1(\Omega))^2 \times (H^1(\Omega) \cap L_0^2(\Omega)),$$

where $\|[\underset{\sim}{v}, q]\|_{0,k} = (\|\underset{\sim}{v}\|_{L^2(\Omega)}^2 + h_k^2\|p\|_{L^2(\Omega)}^2)^{\frac{1}{2}} = ((\underset{\sim}{v}, \underset{\sim}{v})_k + h_k^2(p,p)_k)^{\frac{1}{2}}$. Since the multilevel finite element pressure spaces are nested, we define a prolongation operator as $I_{k-1}^k = [H_{k-1}^k, i_{k-1}^k]$, where i_{k-1}^k is the identity operator on M_{k-1}. Our basic idea of constructing H_{k-1}^k is to define H_{k-1}^k as a composite operator of two single level operators which are defined on two consecutive levels. Such an idea for constructing an intergrid operator is first used for defining two-level Schwarz methods in [4] and [9], and then for defining multigrid methods of plate elements in [10] (but those intergrid operators cannot be expressed in an explicit form). Here, our approach for constructing an operator H_{k-1}^k is to introduce two auxiliary spaces $\underset{\sim}{W}_{k-1}$ and $\underset{\sim}{W}_k$ corresponding to $\underset{\sim}{X}_{k-1}$ and $\underset{\sim}{X}_k$ and satisfying $\underset{\sim}{W}_{k-1} \subset \underset{\sim}{W}_k \subset (C_0(\overline{\Omega}))^2$ and to define $H_{k-1}^k = \beta_k \circ i \circ \alpha_{k-1} = \beta_k \circ \alpha_{k-1}$; α_{k-1}: $\underset{\sim}{X}_{k-1} \to \underset{\sim}{W}_{k-1}$ is an interpolation operator or the modification of an interpolation operator which uses a local averaging technique and $\beta_k : \underset{\sim}{W}_k \to \underset{\sim}{X}_k$ is a interpolation operator. By doing this, there are the following advantages. The first makes the construction of a prolongation operator much easier but the final explicit form of such a prolongation operator is fairly simple. The second reduces the verification of the properties for an intergrid prolongation operator to the verification of similar properties for two single level operators. The third allows us to define several different I_{k-1}^k's. We remark that the convergence analysis can be regarded as a simplification and improvement of [11] by X. Feng and the author. Additional information about multigrid algorithms for solving the mixed finite element equations can be found in [3],[5],[16],[21],[22], where only a few single cases are considered.

An outline of the rest of this paper is as follows. In Section 2, the formation of the prolongation operator I_{k-1}^k and the properties of the operators α_k and β_k are described in detail, and the multigrid algorithm is defined for the mixed finite element approximations of (1.1). In Section 3, the optimal convergence of the multigrid algorithm is demonstrated. Finally, the abstract framework and technique developed in Sections 1-3 is applied to seven typical nonnested mixed finite elements in Section 4. Throughout this paper, unless stated

otherwise, C will denote a generic constant which is independent of the grid level k and mesh size h_k.

2. The prolongation operator and multigrid algorithm. In Sections 2 and 3, we always assume that we are given a family of finite elment spaces $\underset{\sim}{X}_k \times M_k, k \geq 0$ such that $M_{k-1} \subset M_k$, $k \geq 1$ and (1.8)–(1.10) hold. We suppose that there exists a sequence of nested finite element spaces $\{\underset{\sim}{W}_k\}_{k\geq 0}$ associated with $\mathcal{T}_k$, $k \geq 0$, i.e., $\underset{\sim}{W}_{k-1} \subset \underset{\sim}{W}_k \subset (C_0(\overline{\Omega}))^2, k \geq 1$. Also, we assume that two linear operators α_k and β_k exist:

$$\alpha_k : [\underset{\sim}{X}_k + (C_0(\overline{\Omega}))^2](\supset \underset{\sim}{W}_k) \to \underset{\sim}{W}_k, \qquad \beta_k : (\underset{\sim}{W}_k \subset)[(C_0(\overline{\Omega}))^2 + \underset{\sim}{X}_k] \to \underset{\sim}{X}_k. \tag{2.1}$$

We assume that α_k and β_k satisfy the following properties:

$$\alpha_k \circ \alpha_k = \alpha_k, \quad \text{on } \underset{\sim}{W}_k, \qquad \beta_k \circ \beta_k = \beta_k, \quad \text{on } \underset{\sim}{X}_k, \tag{H.1}$$

$$\|\underset{\sim}{v} - \alpha_k \underset{\sim}{v}\|_{L^2(\Omega)} \leq C h_k^2 \|\underset{\sim}{v}\|_{H^2(\Omega)}, \quad \forall\, \underset{\sim}{v} \in (H^2(\Omega) \cap H_0^1(\Omega))^2, \tag{H.2}$$

$$\|\underset{\sim}{v} - \alpha_k \underset{\sim}{v}\|_{L^2(\Omega)} \leq C h_k \|\underset{\sim}{v}\|_k, \qquad \forall\, \underset{\sim}{v} \in \underset{\sim}{X}_k, \tag{H.3}$$

$$\|\underset{\sim}{v} - \beta_k \underset{\sim}{v}\|_{L^2(\Omega)} \leq C h_k^2 \|\underset{\sim}{v}\|_{H^2(\Omega)}, \quad \forall\, \underset{\sim}{v} \in (H^2(\Omega) \cap H_0^1(\Omega))^2, \tag{H.4}$$

$$\|\underset{\sim}{v} - \beta_k \underset{\sim}{v}\|_{L^2(\Omega)} \leq C h_k \|\underset{\sim}{v}\|_k, \qquad \forall\, \underset{\sim}{v} \in \underset{\sim}{W}_k, \tag{H.5}$$

$$\|\alpha_k(\underset{\sim}{v} + \underset{\sim}{w})\|_{L^2(\Omega)} \leq C \|\underset{\sim}{v} + \underset{\sim}{w}\|_{L^2(\Omega)}, \quad \forall\, \underset{\sim}{v} \in \underset{\sim}{X}_k, \quad \underset{\sim}{w} \in \underset{\sim}{W}_k, \tag{H.6}$$

$$\|\beta_k(\underset{\sim}{v} + \underset{\sim}{w})\|_{L^2(\Omega)} \leq C \|\underset{\sim}{v} + \underset{\sim}{w}\|_{L^2(\Omega)}, \quad \forall\, \underset{\sim}{v} \in \underset{\sim}{X}_k, \quad \underset{\sim}{w} \in \underset{\sim}{W}_k. \tag{H.7}$$

We now define the prolongation operator $I_{k-1}^k : \underset{\sim}{X}_{k-1} \times M_{k-1} \to \underset{\sim}{X}_k \times M_k$ as follows:

$$I_{k-1}^k = [H_{k-1}^k, i_{k-1}^k] = [\beta_k \circ i \circ \alpha_{k-1}, i_{k-1}^k] = [\beta_k \circ \alpha_{k-1}, i_{k-1}^k]. \tag{2.2}$$

The relation of H_{k-1}^k, α_{k-1}, and β_k are illustrated by the commutative diagram (2.3):

$$\begin{array}{ccc} \underset{\sim}{X}_k & \xleftarrow{\beta_k} & \underset{\sim}{W}_k \\ {\scriptstyle H_{k-1}^k}\uparrow & & \uparrow{\scriptstyle i} \\ \underset{\sim}{X}_{k-1} & \xrightarrow[\alpha_{k-1}]{} & \underset{\sim}{W}_{k-1} \end{array} \tag{2.3}$$

Following [21], we define the multigrid algorithm for solving the mixed finite element equation at level k as follows. Find $[\underset{\sim}{w}, \rho] \in \underset{\sim}{X}_k \times M_k$ such that

$$\mathcal{L}_k([\underset{\sim}{w}, \rho], [\underset{\sim}{v}, q]) = G_k([\underset{\sim}{v}, q]), \quad \forall\, [\underset{\sim}{v}, q] \in \underset{\sim}{X}_k \times M_k, \tag{2.4}$$

where G_k is a linear functional on $\underset{\sim}{X}_k \times M_k$. In particular, it takes the following form on the finest grid: $G_k([\underset{\sim}{v}, q]) = F_k([\underset{\sim}{v}, q])$.

Multigrid Algorithm

(i) If $k = 0$, (2.4) is solved directly.

(ii) If $k > 0$, let $[\underset{\sim}{w}^0, \rho^0] \in \underset{\sim}{X}_k \times M_k$ be an initial guess and define $[\underset{\sim}{w}^{m+1}, \rho^m] \in \underset{\sim}{X}_k \times M_k$ as follows:

Smoothing step: For $1 \le i \le m, [w^i, \rho^i]$ is defined by

$$(\hat{\underset{\sim}{w}}^i, \underset{\sim}{v})_k + h_k^2(\hat{\rho}^i, q)_k = \Lambda_k{}^{-2}(G_k[\underset{\sim}{v}, q]) - \mathcal{L}_k([\underset{\sim}{w}^{i-1}, \rho^{i-1}], [\underset{\sim}{v}, q]), \ \forall\, [\underset{\sim}{v}, q] \in \underset{\sim}{X}_k \times M_k,$$

$$(\underset{\sim}{w}^i - \underset{\sim}{w}^{i-1}, \underset{\sim}{v})_k + h_k^2(\rho^i - \rho^{i-1}, q)_k = \mathcal{L}_k([\hat{\underset{\sim}{w}}^i, \hat{\rho}^i], [\underset{\sim}{v}, q]), \qquad \forall\, [\underset{\sim}{v}, q] \in \underset{\sim}{X}_k \times M_k.$$

Correction Step: Set

$$[\underset{\sim}{w}^{m+1}, \rho^{m+1}] = [\underset{\sim}{w}^m, \rho^m] + I_{k-1}^k[\underset{\sim}{\psi}, \tau],$$

where $[\underset{\sim}{\psi}, \tau] \in \underset{\sim}{X}_{k-1} \times M_{k-1}$ is the approximation of $[\underset{\sim}{\psi}^*, \tau^*] \in \underset{\sim}{X}_{k-1} \times M_{k-1}$ defined by applying μ iterations with zero as an initial guess of the level $(k-1)$ algorithm to the residual equation

$$\mathcal{L}_{k-1}([\underset{\sim}{\psi}^*, \tau^*], [\underset{\sim}{v}, q]) = G_{k-1}([\underset{\sim}{v}, q]), \quad \forall\, [\underset{\sim}{v}, q] \in \underset{\sim}{X}_{k-1} \times M_{k-1}. \tag{2.5}$$

Here,

$$G_{k-1}([\underset{\sim}{v}, q]) = G_k(I_{k-1}^k[\underset{\sim}{v}, q]) - \mathcal{L}_k([\underset{\sim}{w}^m, \rho^m], I_{k-1}^k[\underset{\sim}{v}, q]).$$

In this algorithm, m is some positive integer to be determined and μ is any positive integer constant greater than or equal to two. In addition, $\Lambda_k = O(h_k^{-2})$ is chosen to be the maximal absolute value of the eigenvalue for the following eigenvalue problem. Find $[\underset{\sim}{\varphi}_k, \nu_k] \in, \underset{\sim}{X}_k \times M_k,\ \lambda \in \mathbb{R} \setminus \{0\}$ such that

$$\mathcal{L}_k([\underset{\sim}{\varphi}_k, \nu_k], [\underset{\sim}{v}, q]) = \lambda((\underset{\sim}{\varphi}_k, \underset{\sim}{v})_k + h_k^2(\nu_k, q)_k), \quad \forall\, [\underset{\sim}{v}, q] \in \underset{\sim}{X}_k \times M_k. \tag{2.6}$$

3. Convergence analysis. In this section, we will discuss the convergence of the algorithm defined in the previous section by using induction. A uniform error reduction rate bounded away from one is proved in the two–grid case provided that sufficiently many smoothing steps are performed. By standard arguments (cf. [2],[12],[14],[18]) the result is then extended to the multilevel algorithm. To show the approximation property, we need to assume H^2–regularity for (1.1), which is true if Ω is a convex polygon (cf. [17]).

Clearly, the eigenvalue problem (2.6) has a complete set of eigenfunctions since $\mathcal{L}_k(\cdot,\cdot)$ is symmetric. Now, let $\{\lambda_j\}$, $\{\underset{\sim}{\varphi}_k^j, \nu_k^j\}$, for $j = 1, 2, \cdots, N_k$, be the eigenvalues and corresponding standard eigenfunctions. Then, for any $[\underset{\sim}{v}_k, q_k] \in \underset{\sim}{X}_k \times M_k$, $c_j,\ j = 1, 2, \cdots, N_k$, exist such that $[\underset{\sim}{v}_k, q_k] = \sum_{j=1}^{N_k} c_j[\underset{\sim}{\varphi}_k^j, \nu_k^j]$. Thus, we define the mesh–dependent norm as follows:

$$|||[\underset{\sim}{v}_k, q_k]|||_{s,k} = \left\{ \sum_{j=1}^{N_k} c_j^2 |\lambda_j|^s \right\}^{\frac{1}{2}} \tag{3.1}$$

It is easy to verify the following inequalities: for $\forall\ [\underset{\sim}{u}_k, p_k],\ [\underset{\sim}{v}_k, q_k] \in \underset{\sim}{X}_k \times M_k$,

$$|||[\underset{\sim}{v}_k, q_k]|||_{0,k} = \|[\underset{\sim}{v}_k, q_k]\|_{0,k}, \tag{3.2}$$

$$|||[\underset{\sim}{v}_k, q_k]|||_{s,k} \le Ch_k^{t-s}|||[\underset{\sim}{v}_k, q_k]|||_{t,k}, \quad t < s, \tag{3.3}$$

$$|\mathcal{L}_k([\underset{\sim}{u}_k, p_k], [\underset{\sim}{v}_k, q_k]) \le |||[\underset{\sim}{u}_k, p_k]|||_{2,k}|||[\underset{\sim}{v}_k, q_k]|||_{0,k}. \tag{3.4}$$

Let $(I_{k-1}^k)^* : \underset{\sim}{X}_k \times M_k \longrightarrow \underset{\sim}{X}_{k-1} \times M_{k-1} (k \ge 1)$ be defined by

$$\mathcal{L}_{k-1}((I_{k-1}^k)^*[\underset{\sim}{v}_k, q_k], [\underset{\sim}{v}_{k-1}, q_{k-1}]) = \mathcal{L}_k([\underset{\sim}{v}_k, q_k], I_{k-1}^k[\underset{\sim}{v}_{k-1}, q_{k-1}]), \tag{3.5}$$
$$\forall\ [\underset{\sim}{v}_{k-1}, q_{k-1}] \in \underset{\sim}{X}_{k-1} \times M_{k-1},\ [\underset{\sim}{v}_k, q_k] \in \underset{\sim}{X}_k \times M_k.$$

Then we have

$$|||(I_{k-1}^k)^*[\underset{\sim}{v}_k, q_k]|||_{2,k-1} \le C|||[\underset{\sim}{v}_k, q_k]|||_{2,k}, \quad \forall\ [\underset{\sim}{v}_k, q_k] \in \underset{\sim}{X}_k \times M_k. \tag{3.6}$$

LEMMA 3.1. *Under the assumptions (H.1)-(H.7), the operator I_{k-1}^k defined by (2.8) and (2.9) satisfies the properties (1.11), (1.12) , i.e., I_{k-1}^k is a "good" prolongation operator.*

It is not difficult to prove Lemma 3.1 by using (H.1)–(H.7) and the triangle inequality. Moreover, by using (1.5)–(1.12), (3.1)–(3.6), and a duality argument similar to that of the proof of Lemma 3.4 (cf. [2],[5],[11],[14],[21]), we can prove the following two lemmas, which, along with Lemma 3.1, are the keys to prove Lemma 3.4 (approximate property).

LEMMA 3.2. *Let $d \in L_0^2(\Omega)$ and $[\underset{\sim}{\sigma}_j, \tau_j] \in \underset{\sim}{X}_j \times M_j$ $(j = k-1, k)$ satisfy*

$$\mathcal{L}_j([\underset{\sim}{\sigma}_j, \tau_j], [\underset{\sim}{v}, q]) = (d, q)_j, \quad \forall\ [\underset{\sim}{v}, q] \in \underset{\sim}{X}_j \times M_j. \tag{3.7}$$

Then we have

$$\|\underset{\sim}{\sigma}_j\|_j + \|\tau_j\|_{L^2(\Omega)} \le C\|d\|_{L^2(\Omega)}, \quad (j = k-1, k), \tag{3.8}$$

$$\|[\underset{\sim}{\sigma}_k, \tau_k] - [\underset{\sim}{\sigma}_{k-1}, \tau_{k-1}]\|_{0,k} \le Ch_k\|d\|_{L^2(\Omega)}. \tag{3.9}$$

LEMMA 3.3. *Let $\underset{\sim}{F} \in L_0^2(\Omega)$ and $[\underset{\sim}{\sigma}_j, \tau_j] \in \underset{\sim}{X}_j \times M_j$ $(j = k-1, k)$ satisfy*

$$\mathcal{L}_j([\underset{\sim}{\sigma}_j, \tau_j], [\underset{\sim}{v}, q]) = (\underset{\sim}{F}, \underset{\sim}{v})_j, \quad \forall\ [\underset{\sim}{v}, q] \in \underset{\sim}{X}_j \times M_j. \tag{3.10}$$

Then we have

$$\|[\underset{\sim}{\sigma}_{k-1}, \tau_{k-1}] - (I_{k-1}^k)^*[\underset{\sim}{\sigma}_k, \tau_k]\|_{0,k-1} \le Ch_k^2\|\underset{\sim}{F}\|_{L^2(\Omega)}. \tag{3.11}$$

We now establish the approximation property.

LEMMA 3.4. (approximation property) *The following inequality holds:*

$$|||[\underset{\sim}{v},q]-I_{k-1}^{k}(I_{k-1}^{k})^{*}[\underset{\sim}{v},q]|||_{0,k}\leq Ch_{k}^{2}|||[\underset{\sim}{v},q]|||_{2,k},\quad\forall\,[\underset{\sim}{v},q]\in\underset{\sim}{X}_{k}\times M_{k}. \tag{3.12}$$

Proof. Let $[\underset{\sim}{\zeta},\theta]=(I_{k-1}^{k})^{*}[\underset{\sim}{v},q]\in\underset{\sim}{X}_{k-1}\times M_{k-1}$, for any $[\underset{\sim}{v},q]\in\underset{\sim}{X}_{k}\times M_{k}$. Then

$$|||[\underset{\sim}{v},q]-I_{k-1}^{k}(I_{k-1}^{k})^{*}[\underset{\sim}{v},q]|||_{0,k}^{2}=\|\underset{\sim}{v}-H_{k-1}^{k}\underset{\sim}{\zeta}\|_{L^{2}(\Omega)}^{2}+h_{k}^{2}\|q-\theta\|_{L^{2}(\Omega)}^{2}. \tag{3.13}$$

By using a duality argument similar to that of [5] and [11], we obtain

$$\|\underset{\sim}{v}-H_{k-1}^{k}\underset{\sim}{\zeta}\|_{L^{2}(\Omega)}\leq Ch_{k}^{2}|||[\underset{\sim}{v},q]|||_{2,k}. \tag{3.14}$$

We now estimate $\|q-\theta\|_{L^{2}(\Omega)}$. Let $[\underset{\sim}{\sigma}_{j},\tau_{j}]\in\underset{\sim}{X}_{j}\times M_{j}$ $(j=k-1,k)$ satisfy

$$\mathcal{L}_{j}([\underset{\sim}{\sigma}_{j},\tau_{j}],[\underset{\sim}{v}',q'])=(q-\theta,q')_{j},\quad\forall\,[\underset{\sim}{v}',q']\in\underset{\sim}{X}_{j}\times M_{j}. \tag{3.15}$$

Then we have

$$\begin{aligned}\|q-\theta\|_{L^{2}(\Omega)}^{2}&=(q-\theta,q)_{k}-(q-\theta,\theta)_{k-1}\\&=\mathcal{L}_{k}([\underset{\sim}{\sigma}_{k},\tau_{k}],[\underset{\sim}{v},q])-\mathcal{L}_{k-1}([\underset{\sim}{\sigma}_{k-1},\tau_{k-1}],[\underset{\sim}{\zeta},\theta])\\&=\mathcal{L}_{k}([\underset{\sim}{\sigma}_{k},\tau_{k}]-I_{k-1}^{k}[\underset{\sim}{\sigma}_{k-1},\tau_{k-1}],[\underset{\sim}{v},q]).\end{aligned} \tag{3.16}$$

Combining (1.11) and (3.4)–(3.11) and using the triangle inequality, we have

$$\|q-\theta\|_{L^{2}(\Omega)}\leq Ch_{k}|||\underset{\sim}{v},q]|||_{2,k}. \tag{3.17}$$

Thus, (3.12) follows from (3.13), (3.14) and (3.17).

Let $[\underset{\sim}{e}^{i},\varepsilon^{i}]=[\underset{\sim}{w}-\underset{\sim}{w}^{i},\alpha-\alpha^{i}]$, $j=0,1,2,...,m+1$, be error functions of the i*th* iteration the multigrid algorithm defined in Section 2 with m smoothing steps at level k. The following smoothing property was proven by Verfürth in [21].

LEMMA 3.5. (smoothing property) *For any initial guess, the following inequality holds:*

$$|||[\underset{\sim}{e}^{m},\varepsilon^{m}]|||_{2,k}\leq Ch_{k}^{-2}m^{\frac{1}{2}}|||[\underset{\sim}{e}^{0},\varepsilon^{0}]|||_{0,k}. \tag{3.18}$$

From the smoothing and approximation properties and by the standard perturbation argument for showing convergence of a W–cycle multigrid algorithm (cf. [2],[12],[14],[18],[21]), we get the following convergence theorem for the multigrid algorithm of Section 2.

CONVERGENCE THEOREM. *Let I_{k-1}^{k} be a "good" prolongation operator and let $\mu>1$ in the multigrid algorithm. Then a constant $0<\gamma<1$ and a positive integer m exist, all independent of the level number k, such that if*

$$|||[\underset{\sim}{\psi}^{*},\tau^{*}]-[\underset{\sim}{\psi},\tau]|||_{0,k}\leq\gamma|||[\underset{\sim}{\psi}^{*},\tau^{*}]|||_{0,k},$$

then

$$|||[\underset{\sim}{w},\rho]-[\underset{\sim}{w}^{m+1},\rho^{m+1}]|||_{0,k}\leq\gamma|||[\underset{\sim}{w},\rho]-[\underset{\sim}{w}^{0},\rho^{0}]|||_{0,k}. \tag{3.19}$$

4. Applications. In this section, we will apply the framework and technique developed and analyzed in the previous sections to seven typical nonnested mixed finite elements for (1.1), which all satisfy

$$M_{k-1}\subset M_k,\quad \underset{\sim}{X}_{k-1}\not\subset \underset{\sim}{X}_k,\quad k\geq 1. \tag{4.1}$$

To do this, we need to construct a sequence of nested auxiliary finite element spaces $\underset{\sim}{W}_k$ satisfying $\underset{\sim}{W}_{k-1}\subset\underset{\sim}{W}_k\subset(C_0(\overline{\Omega}))^2$, to define the operators α_k and β_k, and to give the explicit formulations of the intergrid prolongation operators H_{k-1}^k and $I_{k-1}^k=[H_{k-1}^k,i_{k-1}^k]$ for each specific element. Finally, we need to verify that α_k and β_k satisfy the assumptions (H.1)–(H.7). It then follows from Lemma 3.1 that I_{k-1}^k is a "good" prolongation operator. As has been explained before, we know that β_k and the restrictions of α_k on $(C_0(\overline{\Omega}))^2$ should be some interpolation operators. Therefore, it is quite clear that (H.1), (H.2), and (H.4) hold. Thus, we only need to verify (H.3), (H.5), (H.6), and (H.7). The basic idea for proving these four estimates is to use the fact that a linear operator from a finite dimensional space to another finite dimensional space is bounded, and to combine the standard scaling argument technique(cf. [4],[7],[9],[23]). Here we only give the proof for the Crouzeix–Raviart nonconforming element. The proofs for other elements can be carried out similarly. We hereafter denote by $\overline{\Pi}_k^{\#}$ the local averaging of an interpolation operator $\Pi_k^{\#}$; that means, for any nodal parameter p, $\overline{\Pi}_k^{\#}v(p)=\Pi_k^{\#}v(p)=v(p)$ if v is continuous at p, and $\overline{\Pi}_k^{\#}v(p)$ takes the local average of v at p if v has a jump at p. Finally, we remark that our results show that the bubble function part of the coarse level correction can be ignored in the prolongation step for all elements enriched by bubble functions and that some prolongation operators in existing multigrid algorithms also can be derived by using our technique.

Example 1: *The Mini element and the Bernardi–Raugel element*

These two elements are based on triangles (cf. [1],[6]). Here $\mathcal{T}_k$ is a triangulation of Ω for each $k\geq 0$. The Mini element is defined as follows:

$$\underset{\sim}{X}_k=\{\underset{\sim}{v}\in(C_0(\overline{\Omega}))^2,\ \underset{\sim}{v}|_K\in[P_1(K)\oplus\mathrm{span}\{\lambda_1\lambda_2\lambda_3\}]^2,\ \forall\ K\in\mathcal{T}_k\}, \tag{4.2}$$

$$M_k=\{q\in C(\overline{\Omega})\cap L_0^2(\Omega),\ q|_K\in P_1(K),\ \forall\ K\in\mathcal{T}_k\}. \tag{4.3}$$

The Bernardi–Raugel element is defined as

$$\underset{\sim}{X}_k=\{\underset{\sim}{v}\in(C_0(\overline{\Omega}))^2,\ \underset{\sim}{v}|_K\in[P_1(K)]^2\oplus\mathrm{span}\{\underset{\sim}{p}_1,\underset{\sim}{p}_2,\underset{\sim}{p}_3\},\ \forall\ K\in\mathcal{T}_k\}, \tag{4.4}$$

$$M_k=\{q\in L_0^2(\Omega),\ q|_K\in P_0(K),\ \forall\ K\in\mathcal{T}_k\}, \tag{4.5}$$

where λ_j $(j = 1, 2, 3)$ are the barycentric coordinates and $\underset{\sim}{p}_1 = \lambda_2\lambda_3\underset{\sim}{n}_1$, $\underset{\sim}{p}_2 = \lambda_1\lambda_3\underset{\sim}{n}_2$, $\underset{\sim}{p}_3 = \lambda_1\lambda_2\underset{\sim}{n}_3$, and $\underset{\sim}{n}_j$ $(j = 1, 2, 3)$ are the unit normal vectors of the edges opposite to the vertices a_j $(j = 1, 2, 3)$. It is easy to see that (4.1) holds here.

For both elements, we choose the $\underset{\sim}{W}_k$, α_k, and β_k as follows:

$$\underset{\sim}{W}_k = \{\underset{\sim}{v} \in (C_0(\overline{\Omega}))^2,\ \underset{\sim}{v}|_K \in [P_1(K)]^2, \forall\ K \in \mathcal{T}_k\}, \tag{4.6}$$

$$\alpha_k = \Pi_k^1, \qquad \beta_k = \Pi_k^1, \tag{4.7}$$

where Π_k^1 stands for the linear interpolation operator associated with $\mathcal{T}_k$. Moreover, by using direct computations, we have

$$I_{k-1}^k = [H_{k-1}^k, i_{k-1}^k] = [\beta_k \circ \alpha_{k-1}, i_{k-1}^k] = [\Pi_{k-1}^1, i_{k-1}^k]. \tag{4.8}$$

Now, as it has been explained, we can prove that I_{k-1}^k defined by (4.8) is a "good" prolongation operator. This shows that the *CONVERGENCE THEOREM holds with* I_{k-1}^k *defined by (4.8) for the Mini element and the Bernardi–Raugel element.*

Remark 4.1. For the Mini element, we choose $\underset{\sim}{W}_k$, α_k, and β_k as follows:

$$\underset{\sim}{W}_k = \{\underset{\sim}{v} \in (C_0(\overline{\Omega}))^2,\ \underset{\sim}{v}|_K \in [P_3(K)]^2, \forall\ K \in \mathcal{T}_k\}. \tag{4.9}$$

We define $\alpha_k = \Pi_k^3$, $\beta_k = \Pi_k^m$, where Π_k^3 and Π_k^m stand for the cubic interpolation operator and the Mini element interpolation operator associated with $\mathcal{T}_k$. Then we can get another "good" prolongation operator defined by

$$I_{k-1}^k = [H_{k-1}^k, i_{k-1}^k] = [\beta_k \circ \alpha_{k-1}, i_{k-1}^k] = [\Pi_k^m, i_{k-1}^k].$$

Example 2: *The Crouzeix–Raviart* $P_2^+ - P_1$ *element and the Taylor–Hood* $P_2^+ - P_1$ *element*

They both are triangle elements, which have the same finite element approximate spaces for the velocity field:

$$\underset{\sim}{X}_k = \{\underset{\sim}{v} \in (C_0(\overline{\Omega}))^2,\ \underset{\sim}{v}|_K \in [P_2(K) \oplus \text{span}\{\lambda_1\lambda_2\lambda_3\}]^2,\ \forall\ K \in \mathcal{T}_k\}, \tag{4.10}$$

where λ_j $(j = 1, 2, 3)$ are defined in Example 1. For the Crouzeix–Raviart $P_2^+ - P_1$ element, the finite element space of the pressure field is

$$M_k = \{q \in L_0^2(\Omega),\ q|_K \in P_1(K),\ \forall\ K \in \mathcal{T}_k\}, \tag{4.11}$$

and, for the Taylor–Hood $P_2^+ - P_1$ element,

$$M_k = \{q \in L_0^2(\Omega) \cap C(\overline{\Omega}),\ q|_K \in P_1(K),\ \forall\ K \in \mathcal{T}_k\}. \tag{4.12}$$

Also, it is easy to show that (4.1) holds for these two elements. Here we choose $\underset{\sim}{W}_k$ as in (4.9), i.e., the cubic conforming finite element space on $\mathcal{T}_k$. We define $\alpha_k = \Pi_k^2$, and $\beta_k = \Pi_k^2$ (*or* Π_k^t), where Π_k^2 and Π_k^t stand for the quadratic interpolation operator and the Crouzeix–Raviart $P_2^+ - P_1$ (or the Taylor–Hood $P_2^+ - P_1$) element interpolation operator associated with $\mathcal{T}_k$. Then we have

$$I_{k-1}^k = [H_{k-1}^k, i_{k-1}^k] = [\beta_k \circ \alpha_{k-1}, i_{k-1}^k] = [\Pi_{k-1}^2, i_{k-1}^k]. \tag{4.13}$$

It may be verified that I_{k-1}^k defined by (4.13) is a "good" prolongation operator. Therefore, *the CONVERGENCE THEOREM holds with* I_{k-1}^k *defined by (4.13) for the Crouzeix–Raviart* $P_2^+ - P_1$ *element and the Taylor–Hood* $P_2^+ - P_1$ *element.*

Remark 4.2. For these two elements and the space $\underset{\sim}{W}_k$, we can choose $\alpha_k = \Pi_k^t$ and $\beta_k = \Pi_k^2$, or $\alpha_k = \Pi_k^t$ and $\beta_k = \Pi_k^t$. Then we have $I_{k-1}^k = [H_{k-1}^k, i_{k-1}^k] = [\Pi_k^2, i_{k-1}^k]$ or $I_{k-1}^k = [\Pi_k^t, i_{k-1}^k]$, which are two "good" prolongation operators.

Example 3: *The composite* $P_1 - P_1$ *element*

This is a very simple composite element for (1.1). For each $k \geq 0, \mathcal{T}_k$ is a triangle partition; $\underset{\sim}{X}_k$ and M_k are defined as follows:

$$\underset{\sim}{X}_k = \{\underset{\sim}{v} \in (C_0(\overline{\Omega}))^2,\ \underset{\sim}{v}|_{K_i} \in [P_1(K_i)]^2,\ \overline{K} = \cup_{i=1}^3 \overline{K_i},\ \forall\ K \in \mathcal{T}_k\}, \tag{4.14}$$

$$M_k = \{q \in L_0^2(\Omega) \cap C(\overline{\Omega}),\ q|_K \in P_1(K),\ \forall\ K \in \mathcal{T}_k\}, \tag{4.15}$$

where the K_i $(i = 1, 2, 3)$ are obtained by connecting the three vertices of K with the barycenter. Clearly, this element is stable and satisfies (1.8)–(1.10) (cf. [19]), and (4.1) holds.

Here, we choose $\underset{\sim}{W}_k$ defined by (4.7), i.e., the linear finite element space on $\mathcal{T}_k$, and define $\alpha_k = \Pi_k^1$ and $\beta_k = \Pi_k^1$, where Π_k^1 is defined in Example 1. Therefore, we have

$$I_{k-1}^k = [H_{k-1}^k, i_{k-1}^k] = [\beta_k \circ \alpha_{k-1}, i_{k-1}^k] = [\Pi_k^1, i_{k-1}^k], \tag{4.16}$$

which is a "good" prolongation operator. Thus we have that *the CONVERGENCE THEOREM holds with* I_{k-1}^k *defined by (4.16) for the composite* $P_1 - P_1$ *element.*

Example 4: *The Crouzeix–Raviart nonconforming element*

This is the most well–known nonconforming finite element (cf. [8]). For each $k \geq 0, \mathcal{T}_k$ is a triangle partition; $\underset{\sim}{X}_k, M_k$ are defined by

$$\underset{\sim}{X}_k = \{\underset{\sim}{v}, \underset{\sim}{v}|_K \in [P_1(K)]^2, \forall K \in \mathcal{T}_k, \underset{\sim}{v} \text{ is continuous at } p \in \mathcal{N}_k, \underset{\sim}{v}(p) = 0, p \in \partial\mathcal{N}_k\}, \tag{4.17}$$

$$M_k = \{q \in L_0^2(\Omega),\ q|_K \in P_0(K), \forall\ K \in \mathcal{T}_k\}. \tag{4.18}$$

where and in the next example, $\mathcal{N}_k$ stands for the set of midpoints of the edges of $\mathcal{T}_k$ in Ω, and $\partial\mathcal{N}_k$ is the set of midpoints along $\partial\Omega$. Obviously, (4.1) holds for the Crouzeix–Raviart nonconforming element. Here we define the $\underset{\sim}{W}_k$, α_k, and β_k as follows:

$$\underset{\sim}{W}_k = \{\underset{\sim}{v} \in (C_0(\overline{\Omega}))^2,\ \underset{\sim}{v}|_e \in [P_2(e)]^2, \forall\ e \in \mathcal{T}_{k+1}\}, \tag{4.19}$$

$$\alpha_k = \overline{\Pi}^2_{k+1}, \qquad \beta_k = \Pi^c_k, \tag{4.20}$$

where Π^2_k is defined in Example 2 and Π^c_k stands for the standard Crouzeix–Raviart nonconforming element interpolation operator associated with $\mathcal{T}_k$. Then we obtain after some computations

$$I^k_{k-1} = [H^k_{k-1}, i^k_{k-1}] = [\beta_k \circ \alpha_{k-1}, i^k_{k-1}] = [\overline{\Pi}^c_k, i^k_{k-1}]. \tag{4.21}$$

Moreover, the following two Lemmas will show that I^k_{k+1} defined by (4.21) is a "good" prolongation operator. Hence, *the CONVERGENCE THEOREM holds with I^k_{k-1} defined by (4.21) for the Crouzeix–Raviart nonconforming element.*

LEMMA 4.1. *α_k and β_k defined by (4.20) satisfy (H.3) and (H.5), respectively.*

Proof. We only give a proof for (H.3). Similarly, (H.5) can be carried out. For any $K \in \mathcal{T}_k$, we denote a domain $G(K) \subset \Omega$ such that $\overline{G(K)} = \cup\{\overline{K'} \in \mathcal{T}_k,\ \overline{K'} \cap \overline{K} \neq \emptyset\}$, and let

$$\alpha_{G(K)} = \alpha_k|_{G(K)} : \underset{\sim}{X}_k|_{G(K)} \to \underset{\sim}{W}_k|_K.$$

Thus, if $\partial K \cap \partial\Omega = \emptyset$, then it is not hard to show that $P_0(G(K)) \subset \underset{\sim}{X}_k|_{G(K)}$, $\alpha_{G(K)}$ is a linear operator, and

$$\alpha_{G(K)} p_0 = p_0, \quad \forall p_0 \in P_0(G(K)).$$

Furthermore, $\|\nabla \underset{\sim}{v}\|$ is a norm over $\underset{\sim}{X}_k|_{G(K)}$. Therefore, for any $K \in \mathcal{T}_k$ with $\partial K \cap \partial\Omega = \emptyset$, we have, by using the standard scaling argument,

$$\|\underset{\sim}{v} - \alpha_{G(K)} \underset{\sim}{v}\|_{L^2(K)} \leq C h_k \|\nabla \underset{\sim}{v}\|_{L^2(G(K))} \quad \forall\ \underset{\sim}{v} \in \underset{\sim}{X}_k|_{G(K)}. \tag{4.22}$$

For any $K \in \mathcal{T}_k$ with $\partial K \cap \partial\Omega \neq \emptyset$, $\|\nabla \underset{\sim}{v}\|$ is still a norm over $\underset{\sim}{X}_k|_{G(K)}$ since $\underset{\sim}{v}$ vanishes at all midpoints of the sides along $\partial\Omega$. Therefore, (4.22) still holds. Hence, (H.3) follows from summing up (4.22) for all $K \in \mathcal{T}_k$.

LEMMA 4.2. *α_k and β_k defined by (4.20) satisfy (H.6) and (H.7), respectively.*

Proof. We only consider (H.7). Similarly, (H.6) can be treated. For any $K \in \mathcal{T}_k$, it is easy to see that

$$\beta_K = \beta_k|_K : (\underset{\sim}{W}_k + \underset{\sim}{X}_k)|_K \to \underset{\sim}{X}_k|_K$$

is a linear operator. So, by using the standard scaling argument technique, we have that

$$\|\beta_K(v+w)\| \leq C\|(v+w)\|, \quad \forall v \in \underset{\sim}{X}_k|_K, \quad w \in \underset{\sim}{W}_k|_K. \tag{4.23}$$

Thus, summing up (4.23) for all $K \in \mathcal{T}_k$, we complete the proof for (H.7).

Remark 4.3. If we choose

$$\underset{\sim}{W}_k = \{\underset{\sim}{v} \in (C_0(\overline{\Omega}))^2,\ \underset{\sim}{v}|_K \in [P_1(K)]^2, \forall\ K \in \mathcal{T}_k\},$$

$\alpha_k = \overline{\Pi}_k^1$, and $\beta_k = \Pi_k^1$, where Π_k^1 is defined in Example 1, then

$$I_{k-1}^k = [H_{k-1}^k, i_{k-1}^k] = [\beta_k \circ \alpha_{k-1}, i_{k-1}^k]$$

is also a "good" prolongation operator. Again if we choose

$$\underset{\sim}{W}_k = \{\underset{\sim}{v} \in (C_0(\overline{\Omega}))^2,\ \underset{\sim}{v}|_e \in [P_1(e)]^2, \forall\ e \in \mathcal{T}_{k+1}\},$$

$\alpha_k = \overline{\Pi}_{k+1}^1$, and $\beta_k = \overline{\Pi}_k^c$, then we have a "good" prolongation operator again.

Example 5: *A rectangular nonconforming element*

This element was proposed in [15]. It can be regarded as an extension of the Crouzeix-Raviart nonconforming element to the case of a rectangle. This element has five element nodal parameters, which are the function values at the four midpoints of the sides and the center of the rectangle. For each $k \geq 0, \mathcal{T}_k$ is a rectangle partition, $\underset{\sim}{X}_k$ and M_k are defined as follows:

$$(4.24)\quad \underset{\sim}{X}_k = \{\underset{\sim}{v}, \underset{\sim}{v}|_K \in P_K, \forall K \in \mathcal{T}_k, \underset{\sim}{v} \text{ is continuous at } p \in \mathcal{N}_k, \underset{\sim}{v}(p) = 0, \forall\ p \in \partial\mathcal{N}_k\},$$

$$(4.25)\quad M_k = \{q \in L_0^2(\Omega),\ q|_k \in Q_0(K), \forall\ K \in \mathcal{T}_k\},$$

where

$$P_K = \{\ p(x) = \hat{p}(F_K^{-1}(x)),\ \ \hat{p} \in \hat{P}\}.$$

Here F_K is an affine mapping from the rectangle K to the reference rectangle $\hat{K}$, and $\hat{P} = \text{span}\{1, x_1, x_2, \varphi(x_1), \varphi(x_2)\ \text{ on }\ \hat{K}\}$, $\varphi(t) = \frac{1}{2}(5t^4 - 3t^2)$. Then, (4.1) holds for this rectangular nonconforming element.

For this rectangular nonconforming element, we take $\underset{\sim}{W}_k$ as follows:

$$(4.19)\qquad \underset{\sim}{W}_k = \{\underset{\sim}{v} \in (C_0(\overline{\Omega}))^2,\ \underset{\sim}{v}|_e \in [Q_2(e)]^2, \forall\ e \in \mathcal{T}_{k+1}\},$$

$\alpha_k = \overline{\Pi}_{k+1}''$ is defined as in Example 4, and $\beta_k = \Pi_k^*$, which is the standard rectangular nonconforming element interpolation operator associated with $\mathcal{T}_k$. We then obtain

$$(4.31)\qquad I_{k-1}^k = [H_{k-1}^k, i_{k-1}^k] = [\beta_k \circ \alpha_{k-1}, i_{k-1}^k] = [\overline{\Pi}_k^*, i_{k-1}^k].$$

We can show that this I_{k-1}^k is a "good" prolongation operator. Therefore, *the CONVERGENCE THEOREM holds with I_{k-1}^k defined by (4.31) for this rectangular nonconforming element.*

REFERENCES

[1] D. N. ARNOLD, F. BREZZI AND M. FORTIN, *A stable finite element for the Stokes equations*, Calcolo, 21 (1984), pp. 337–344.

[2] R. E. BANK AND T.DUPONT, *An optimal order process for solving finite element equations*, Math. Comp., 36 (1980), pp. 35–51.

[3] D. BRAESS AND R. VERFÜRTH, *Multigrid methods for nonconforming finite element methods*, SIAM J. Numer. Anal., 27 (1988), pp. 979–986.

[4] S. C. BRENNER, *Two-level additive Schwarz preconditioners for nonconforming finite element method*, preprint.

[5] ———, *A nonconforming mixed multigrid method for the pure displacement problem in planar linear elasticity*, SIAM J. Numer. Anal., 30 (1993), pp. 116–135.

[6] F. BREZZI AND M. FORTIN, *Mixed and Hybrid Finite Element Methods*, Springer–Verlag, New York, 1991.

[7] P. G. CIARLET, *The Finite Element Method for Elliptic Problems*, North–Holland, Amsterdam, 1978.

[8] M. CROUZEIX AND P.A. RAVIART, *Conforming and nonconforming finite element methods for solving the stationary Stokes equations I*, R.A.I.R.O. Modél. Math. Anal. Numér., R-3 (1973), pp. 33–75.

[9] Q. DENG AND X. FENG, *Schwarz methods for conforming and nonconforming finite element approximations of plate bending problems, Part I: Two-level methods*, Technical Report MA-01-94, The University of Tennessee, 1994.

[10] ———, *Optimal order nonnested multigrid methods for the biharmonic problems*, preprint.

[11] ———, *Multigrid methods for the Stokes equations by mixed methods*, preprint.

[12] C. DOUGLAS, *Multigrid algorithms with applications to elliptic boundary–value problems*, SIAM J. Numer. Anal., 21 (1984), pp. 236–254.

[13] V. GIRAULT AND P. A. RAVIART, *Finite Element Methods for Navier–Stokes Equations*, Springer –Verlag, Berlin and Heidelberg, 1986.

[14] W. HACKBUSCH, *Multigrid Methods and Applications*, Springer–Verlag, Heidelberg and New York, 1985.

[15] H. HAN, *Nonconforming element in the mixed finite element method*, J. Comput. Math., 3 (1984), pp. 223–233.

[16] Z. HUANG, *A multi–grid algorithm for mixed problems with penalty*, Numer. Math., 57 (1990), pp. 227–247.

[17] R. B. KELLOGG AND J. E. OSBORN, *A regularity result for the Stokes problem on a convex polygon*, J. Funct. Anal., 21 (1976), pp. 397–431.

[18] J. MANDEL, S. MCCORMICK, AND R. BANK, *Variational multigrid theory*, in Multigrid Methods, S. McCormick, ed., SIAM Frontiers in Applied Mathematics, Vol. 3, SIAM, Philadelphia, 1987, pp. 131–177.

[19] J. QIN, *On the Convergence of Some Low Order Mixed Finite Elements for Incompressible Fluids*, Ph.D. thesis, Depart. of Math., Penn. State Univ., 1994.

[20] R. TEMAM, *Navier–Stokes Equations, Theory and Numerical Analysis*, North Holland, Amsterdam, 1983.

[21] R. VERFÜRTH, *A multi–level algorithm for mixed problems*, SIAM J. Numer. Anal., 21 (1984), pp. 264–271.

[22] ———, *Multi–level algorithms for mixed problems II. Treatment of the mini–element*, SIAM J. Numer. Anal., 25 (1988), pp. 285–293.

[23] J. XU, *Theory of Multilevel Methods*, Ph.D. Thesis, Dept. of Math., Cornell Univ., 1989.

Page intentionally left blank

A NOTE ON MULTIGRID THEORY FOR NON-NESTED GRIDS AND/OR QUADRATURE

C. C. Douglas
IBM Thomas J. Watson Research Center
Yorktown Heights, NY
and
Department of Computer Science, Yale University
New Haven, CT

J. Douglas, Jr.
Department of Mathematics, Purdue University
West Lafayette, IN

D. E. Fyfe
Laboratory for Computational Physics and Fluid Dynamics
Naval Research Laboratory
Washington, DC

SUMMARY

We provide a unified theory for multilevel and multigrid methods when the usual assumptions are not present. For example, we do not assume that the solution spaces or the grids are nested. Further, we do not assume that there is an algebraic relationship between the linear algebra problems on different levels. What we provide is a computationally useful theory for adaptively changing levels. Theory is provided for multilevel correction schemes, nested iteration schemes, and one way (i.e., coarse to fine grid with no correction iterations) schemes. We include examples showing the applicability of this theory: finite element examples using quadrature in the matrix assembly and finite volume examples with non-nested grids. Our theory applies directly to other discretizations as well.

INTRODUCTION

In this paper, we do not make the usual multigrid assumptions. In particular, the grids are not necessarily nested. The norms correspond to inner products on a grid,

but the inner products are not necessarily identical from level to level. There may or may not be algebraic relationship between the linear algebra problems on different levels.

We provide what is really three level analysis rather than the more traditional two level theory. Among other things, this provides a rigorous basis for adaptively changing levels.

Assume that there are j spaces $\mathcal{M}_k$, $1 \leq k \leq j$, approximating some solution space $\mathcal{M}$. Also assume that $\dim \mathcal{M}_k \leq \dim \mathcal{M}_{k+1}$.

A set of approximate problems

$$A_k u_k + f_k = 0, \quad u_k, f_k \in \mathcal{M}_k, \quad A_k \in \mathcal{L}(\mathcal{M}_k), \tag{1}$$

will be solved approximately instead of the desired linear problem

$$Au + f = 0, \quad u, f \in \mathcal{M}, \quad A \in \mathcal{L}(\mathcal{M}).$$

As usual in multigrid procedures, two sets of mappings between neighboring spaces are assumed to exist: The prolongation (or interpolation) mappings are

$$\mathcal{P}_{k-1} : \mathcal{M}_{k-1} \to \mathcal{M}_k \qquad \text{prolongation (or interpolation)}$$

$$\mathcal{R}_k : \mathcal{M}_k \to \mathcal{M}_{k-1} \qquad \text{restriction (or projection)}$$

In some cases, each A_k is related to A_{k+1} by

$$A_k = \mathcal{R}_{k+1} A_{k+1} \mathcal{P}_k.$$

However, the theorems in this paper do not assume this relation.

For partial differential equations that are discretized in a standard fashion, there can be natural definitions for $\mathcal{R}_{k+1}$ and $\mathcal{P}_k$. Some of these are described in detail and shown graphically in [1] and [5].

Now, define a k-level *standard* correction multilevel algorithm:

ALGORITHM MG(k, $\{\mu_\ell\}_{\ell=1}^{j}$, x_k, f_k)
(1) If $k = 1$ or $\mu_k = 0$, then solve $A_k x_k + f_k = 0$ to some accuracy.
(2) If $k > 1$ and $\mu_k > 0$, then repeat (2a)–(2c) for $i = 1, \cdots, \mu_k$:
  (2a) Update x_k using the pre-solver.
  (2b) Solve a residual correction problem on level $k-1$:
    $x_k \leftarrow x_k + \mathcal{P}_{k-1}$ MG($k-1$, $\{\mu_\ell\}_{\ell=1}^{j}$, 0, $\mathcal{R}_k(A_k x_k + f_k)$).
  (2c) Update x_k using the post-solver.
(3) Return x_k.

It is assumed that $0 \leq \mu_1, \mu_j \leq 1$ in this definition. In practice, $\mu_j > 1$ is common, but this can be interpreted as the repetition μ_j times of the algorithm for the case of $\mu_j = 1$.

On all but level 1 (the coarsest grid level), two solvers are associated with a level: a pre-solver and a post-solver. These surround the coarser level correction (2b). In most real applications, only one solver is associated with a level (one of the pre- or post-solvers is the identity operation). The solvers can be smoothers, roughers, or direct solvers.

In order to analyze Algorithm MG from an iterative method viewpoint, we transform it into a *nonstandard* form similar to that introduced in [1]. First, add an additional level $j+1$, which is just a repetition of level j:

$$\mathcal{M}_{j+1} = \mathcal{M}_j, \quad \mathcal{P}_j = \mathcal{R}_{j+1} = I, \quad A_{j+1} = A_j, \quad C_{1,j} = C_{2,j} = 1.$$

The initial residual z_{j+1} is then given by

$$A_{j+1}x_{j+1} + f = z_{j+1}.$$

All analysis can now be done using residual correction problems.

Define the following:

z_{k+1}	The residual on level $k+1$ at some step.
$x_k^{(-1)}$	The initial guess for level k; this is normally 0, except on level $j+1$.

Now, define a k-level *nonstandard* correction multilevel algorithm:

ALGORITHM NSMG(k, $\{\mu_\ell\}_{\ell=1}^{j}$, z_{k+1}, $x_k^{(-1)}$)

(1) Initial residual: $\mathcal{R}_{k+1}z_{k+1} \in \mathcal{M}_k$.

(2) Initial pre-solve: Update $x_k^{(-1)}$ to get $x_k^{(0)}$ such that
$A_k x_k^{(0)} + \mathcal{R}_{k+1}z_{k+1} = z_k^{(0)}$, where $\|z_k^{(0)}\| \leq \rho_k^{(1)}\|z_{k+1}\|$.

(3) Let $\hat{x}_k^{(1)} = x_k^{(0)}$, $\hat{z}_k^{(1)} = z_k^{(0)}$, and $\gamma_1^{(1)} = 0$.

(4) If $\mu_k > 0$, then repeat $i = 1, \cdots, \mu_k$:

(4a) If $i > 1$, then

(4a1) Residual: $A_k x_k^{(i-1)} + \mathcal{R}_{k+1}z_{k+1} = \hat{\theta}_k^{(i)}$.

(4a2) Pre-solve: Update $x_k^{(i-1)}$ to get $\hat{x}_k^{(i)}$ such that
$A_k\hat{x}_k^{(i)} + \mathcal{R}_{k+1}z_{k+1} = \hat{z}_k^{(i)}$, where $\|\hat{z}_k^{(i)}\| \leq \rho_k^{(i)}\|\hat{\theta}_k^{(i)}\|$.

(4b) If $k > 1$, then

(4b1) Correction: $\gamma_k^{(i)} = \mathcal{P}_{k-1}\bar{x}_{k-1}^{(i)}$, where
$\bar{x}_{k-1}^{(i)} = \text{NSMG}(k-1, \{\mu_\ell\}_{\ell=1}^{j}, \hat{z}_k^{(i)}, 0)$ and
$A_{k-1}\bar{x}_{k-1}^{(i)} + \mathcal{R}_k\hat{z}_k^{(i)} = \bar{z}_{k-1}^{(i)}$.

(4c) Calculate $\sigma_k^{(i)}$:
$$\|\hat{z}_k^{(i)} + A_k \mathcal{P}_{k-1} \bar{x}_{k-1}^{(i)}\| \leq \sigma_k^{(i)} \|\hat{z}_k^{(i)} + \mathcal{P}_{k-1} A_{k-1} \bar{x}_{k-1}^{(i)}\|.$$
(4d) Residual: $A_k(\hat{x}_k^{(i)} + \gamma_k^{(i)}) + \mathcal{R}_{k+1} z_{k+1} = \theta_k^{(i)}$.
(4e) Post-solve: Update $\hat{x}_k^{(i)} + \gamma_k^{(i)}$ to get $x_k^{(i)}$ such that
$$A_k x_k^{(i)} + \mathcal{R}_{k+1} z_{k+1} = z_k^{(i)}, \text{ where } \|z_k^{(i)}\| \leq \epsilon_k^{(i)} \|\theta_k^{(i)}\|.$$
(5) Return $x_k^{(\mu_k)}$.

This is almost the same algorithm as was analyzed in [1]. The difference is in step (4c). Here we calculate the norm of the difference between the effect of two similar operators on the correction with respect to the residual before the correction was computed.

Consider the example of adaptively changing levels based on reducing the residual norm adequately. We can calculate $\sigma_k^{(i)}$ while computing a correction in step (4b). Based on the size of $\sigma_k^{(i)}$, we can determine if the current candidate for $\bar{x}_{k-1}^{(i)}$ is sufficient in order to maintain convergence on level k (or a fast enough convergence rate). Should $\sigma_k^{(i)}$ be too large, more corrections on level $k-2$ or a better approximation on level $k-1$ might be appropriate.

In order to consider a priori analysis, the actual forms for $\rho_k^{(i)}$ and $\epsilon_k^{(i)}$ should be substituted. Examples of these forms can be found for various elliptic partial differential equations and iterative solvers in [2] and [3].

A second multigrid variant is a nested iteration scheme, which begins computation on level 1 and traverses the levels to some level j, using each level k, $k < j$, to generate an initial guess for level $k+1$ and possibly for solving residual correction problems. Define a k-level *standard* nested iteration multigrid scheme by

ALGORITHM NI$(j, \{\mu_k\}_{k=1}^j, x_1, \{f_k\}_{k=1}^j)$
(1) For $k = 1, \cdots, j$, do
($1a$) If $k > 1$, then $x_k \leftarrow \mathcal{P}_{k-1} x_{k-1}$
($1b$) $x_k \leftarrow$MG$(k, \{\mu_\ell\}_{\ell=1}^k, x_k, f_k)$
(2) Return x_j

Note that $\mu_\ell = 1$, all ℓ, corresponds to full multigrid (or nested iteration V cycle). Choosing $\mu_\ell = 0$, all ℓ, corresponds to one way multigrid, i.e., no correction cycles whatsoever (see [3] and [4]).

Define a *nonstandard* nested iteration multilevel algorithm by

ALGORITHM NSNI(j, $\{\mu_\ell\}_{\ell=1}^{j}$, $x_1^{(-1)}$)

(1) repeat $k = 1, \cdots, j$:

  (1a) Initial guess: If $k > 1$, then

    (1a1) $x_k^{(-1)} = \mathcal{P}_{k-1} x_{k-1}^{(\mu_{k-1})}$.

  (1b) Residual: $z_k = A_k x_k^{(-1)} + f_k$.

  (1c) Solve: $x_k^{(\mu_k)}$ =NSMG(k, $\{\mu_\ell\}_{\ell=1}^{j}$, z_k, $x_k^{(-1)}$).

(2) Return $x_j^{(\mu_j)}$.

THEORY

In this section, we state some basic theorems, based on a simple theory that is computationally useful, including for adaptively changing levels. See [5] for the proofs.

Associated with each level is a norm, $\|\cdot\|_k$. Assume that

$$C_{1,k}\|u\|_k \le \|\mathcal{P}_k u\|_{k+1} \le C_{2,k}\|u\|_k, \quad \forall u \in \mathcal{M}_k,$$

where the forms of $C_{1,k}$ and $C_{2,k}$ are known; these constants can depend on the coefficients in the differential problem and on the grid. A large value of $C_{2,k}$ will inhibit the rate of convergence.

The basic theorem for Algorithm NSMG is the following:

Theorem 1. *Assume the following for all levels $1 \le k \le j$:*

1. *z_{j+1} is the residual on level $j+1 \ge 2$.*
2. *$z_k^{(i)}$ is the residual on level k at step i.*
3. *$\|\hat{z}_k^{(i)} + A_k \mathcal{P}_{k-1} \bar{x}_{k-1}^{(i)}\| \le \sigma_k^{(i)} \|\hat{z}_k^{(i)} + \mathcal{P}_{k-1} A_{k-1} \bar{x}_{k-1}^{(i)}\|$.*
4. *$\|(I - \mathcal{P}_{k-1}\mathcal{R}_k) z_k^{(i)}\| \le \delta_k^{(i)} \|z_k^{(i)}\|$.*

Let

$$E_1^{(1)} = \epsilon_1^{(1)} \rho_1^{(1)} \quad \textit{and} \quad E_k^{(\mu_k)} = \prod_{i=1}^{\mu_k} \left(\epsilon_k^{(i)} \rho_k^{(i)} \sigma_k^{(i)} \left[\delta_k^{(i)} + C_{2,k-1} E_{k-1}^{(\mu_{k-1})} \right] \right).$$

Then,

$$\|z_k^{(\mu_k)}\|_k \le E_k^{(\mu_k)} \|z_{k+1}\|_{k+1}.$$

The proof of Theorem 1 is a double induction argument and can be found in [5].

A more precise analysis, based on an affine space decomposition of each $\mathcal{M}_k$, would follow the analysis in [1].

The basic theorem for Algorithm NSNI is the following:

Theorem 2. *Make the same assumptions as in Theorem 1. Further assume that (1) is approximated by some ξ_k such that*

$$A_k\xi_k + f_k = \theta_k \tag{2}$$

starting from some initial guess $x_k = \mathcal{P}_{k-1}\xi_{k-1}$. Given some $\{\zeta_k\}_{k=1}^{j}$, we want

$$\begin{cases} \|\theta_1\| & \leq & \zeta_1\|A_1x_1 + f_1\|, \\ \|\theta_k\| & \leq & \zeta_k\|\theta_{k-1}\|, \quad 1 < k \leq j. \end{cases}$$

Then

$$E_k^{(\mu_k)} \leq \zeta_k \quad \text{for } 1 \leq k \leq j \tag{3}$$

for an appropriate choice of $\{\{\rho_k^{(i)}, \epsilon_k^{(i)}\}_{i=1}^{\mu_k}\}_{k=1}^{j}$.

The proof of Theorem 2 is obvious (see [3] for example). Note that by calculating $\delta_k^{(i)}$ and $\sigma_k^{(i)}$ as a computation progresses, the choice of $\rho_k^{(i)}$ and $\epsilon_k^{(i)}$ can be chosen adaptively to ensure that (3) is satisfied.

The one way multigrid method is a common computational method in engineering applications. It has been used for decades as a method for producing an initial guess on the grid in which a solution to a problem is actually wanted. This process is described in [4] for a procedure that he first saw in the 1920's.

Consider a typical partial differential equation problem to be solved numerically. It is discretized on a set of grids Ω_k, $1 \leq k \leq j$, with some notion of grid spacing (or a mesh diameter) h_k.

The basic theorem for one way multigrid is the following:

Theorem 3. *Make the same assumptions as in Theorem 2. Further assume $\mu_k = 0$, $1 \leq k \leq j$, and that*

$$\theta_k = Ch_k^q, \quad C, q, h > 0 \in \mathbb{R}.$$

Then

$$\zeta_k = CC_{2,k}(h_k/h_{k-1})^q$$

is adequate to ensure that (2) is satisfied. Hence, (3) is satisfied with $\rho_k^{(1)} = \zeta_k$.

Once again the proof is obvious. Note Theorem 3 gives a simple bound for one way multigrid that is independent of the solver used on each level.

FINITE VOLUME EXAMPLE

Consider the two-point boundary value problem

$$\begin{cases} -(a(x)u_x)_x + c(x)u & = & f(x), \qquad x \in \Omega = [0,1], \\ u(0) = u(1) & = & 0. \end{cases} \tag{4}$$

A finite volume discretization of (4) yields

$$a_{i-1/2}u_{i-1}^k + (\Delta x_i c_i - a_{i-1/2} - a_{i+1/2})u_i^k + a_{i+1/2}u_{i+1}^k = \Delta x_i f_i, \quad i = 1, \ldots, N$$

on level k where $a_{i+1/2} = 2A_{i+1/2}/(\Delta x_i + \Delta x_{i+1})$, and Δx_i is the length of cell (interval) i. While the grid points $x_{i+1/2}$ in a finite volume multigrid procedure are nested, the locations of the unknowns u are not nested.

Clearly, one would not use a multigrid approach to solve this problem. However, multigrid is a viable alternative for the equivalent multi-dimensional problem. The following remarks generalize to the multi-dimensional case through the use of tensor product formulations for the prolongation and restriction operators. We discuss the one-dimensional case for clarity.

Let us define a restriction matrix

$$\mathcal{R}_k = \frac{1}{2}\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & \ldots & \ldots & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & \ldots & \ldots & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & \ldots & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & & & \ddots & \ddots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \ldots & \ldots & 1 & 1 \end{pmatrix}.$$

This is just piecewise linear interpolation. We can also define a prolongation matrix $\mathcal{P}_{k-1} = 2\mathcal{R}_k{}^T$. This prolongation matrix corresponds to piecewise constant interpolation; clearly, not a very accurate choice, but a demonstrative one.

If we formulate the coarse grid matrix from $A_{k-1} = \mathcal{R}_k A_k \mathcal{P}_{k-1}$ and a restricted right-hand-side from $\mathcal{R}_k$, we obtain

$$a_{2i-3/2}u_{i-1}^{k-1} + (\Delta x_{2i}c_{2i} + \Delta x_{2i-1}c_{2i-1} - a_{2i-3/2} - a_{2i+1/2})u_i^{k-1} +$$

$$a_{2i+1/2}u_{i+1}^{k-1} = \Delta x_{2i-1}f_{2i-1} + \Delta x_{2i}f_{2i}, \quad i = 1, \ldots, N/2.$$

This is a reasonable coarse grid approximation where the only difference from the finite volume discretization on the coarse grid would be in the use of the underlying fine grid to discretize the finite volume integral.

A straightforward calculation of $\|(I - \mathcal{P}_{k-1}\mathcal{R}_k)x)\|$ for arbitrary x shows that $\delta_k = 1/\sqrt{2}$.

A more practical prolongation matrix would use quadratic interpolation (see [5]). The use of this prolongation matrix in the definition of the coarse matrices would expand the bandwidth of each successive coarser matrix. This defeats the purpose of multigrid where one expects to do less work on the coarser grids. The use of this prolongation matrix with the piecewise linear interpolation restriction matrix gives $\delta_k = \sqrt{531}/32$.

We can calculate δ_k for multidimensional problems when tensor product meshes are in use. The calculation of δ_k for the piecewise linear–piecewise constant case is

fairly easy. In this case, the $\mathcal{P}_{k-1}\mathcal{R}_k$ calculation in one dimension separates into a local computation. It effectively collects the cells on the fine grid pairwise to form the coarse cell by averaging. In multiple dimensions, because of the tensor product nature of $\mathcal{P}_{k-1}$ and $\mathcal{R}_k$, that still happens. Hence, the algebra produces a worst case δ_k of

$$\sqrt{(2^d - 1)/2^d},$$

where d is the dimension of the problem. So,

d	δ_k
1	$\sqrt{1/2}$
2	$\sqrt{3/4}$
3	$\sqrt{7/8}$

This approaches 1 rapidly. However, in any given application, δ_k can be smaller.

The quadratic interpolation case provides better results in multiple dimensions, as would be expected.

AN EXAMPLE OF THE FAILURE OF $A_k = \mathcal{R}_{k+1}A_{k+1}\mathcal{P}_k$

Consider the boundary value problem

$$\begin{cases} -\sum_{i,j=1}^{2}(a_{ij}(x)u_{x_i})_{x_j} + b_i(x)u_{x_i} + c(x)u = f(x), & x \in \Omega = [0,1]^2, \\ u = 0, & x \in \partial\Omega. \end{cases} \tag{5}$$

Let the k-level partition $\mathcal{S}_k$ of Ω consist of squares of side length $2^{-(k+\ell)}$, where ℓ is independent of k. Let the k-level finite element space $\mathcal{M}_k$ consist of C^0-bilinear functions over $\mathcal{S}_k$ that vanish on $\partial\Omega$. Then, the natural k-level Galerkin equations,

$$A_k u_k = \phi_k,$$

would be generated by seeking a function $\tilde{u}^k \in \mathcal{M}_k$ such that

$$\sum_{i,j=1}^{2}(a_{ij}\tilde{u}^k_{x_i}, \tilde{v}^k_{x_j}) + \sum_{i=1}^{2}(b_i\tilde{u}^k_{x_i}, \tilde{v}^k) + (c\tilde{u}^k, \tilde{v}^k) = (f, \tilde{v}^k), \quad \tilde{v}^k \in \mathcal{M}_k, \tag{6}$$

where $(\cdot,\cdot)$ indicates the inner product on $L^2(\Omega)$. Note that exact integration is not, in general, feasible. Thus, it is usually necessary to invoke a quadrature rule to approximate the integrals in (6). A (2×2)-Gauss quadrature rule suffices to maintain unique solvability of the resulting linear equations, along with the proper asymptotic order of accuracy of the k-level approximation to the solution of (5). Denote by $(\cdot,\cdot)_G$

the (2×2)-Gauss quadrature approximation to $(\cdot,\cdot)$. Define the k-level equations (5) through the approximation

$$\sum_{i,j=1}^{2} (a_{ij}\tilde{u}^k_{x_i}, \tilde{v}^k_{x_j})_G + \sum_{i=1}^{2} (b_i \tilde{u}^k_{x_i}, \tilde{v}^k)_G + (c\tilde{u}^k, \tilde{v}^k)_G = (f, \tilde{v}^k)_G, \quad \tilde{v}^k \in \mathcal{M}_k.$$

Consider the feasibility of the relation $A_k = \mathcal{R}_{k+1} A_{k+1} \mathcal{P}_k$ by making a simple parameter count. If the prolongation and restriction operators are defined in terms of the parameters related to the vertex values of a single element in the coarser partition and the vertex values of the corresponding four squares in the finer partition, it suffices to consider a unit square S^1 for the coarser element (associated with index 1) and its partition (associated with index 2) into four squares, S^2_j, $j = 1, \ldots, 4$, for the finer elements. Note that the sixteen quadrature points on S^2_j are distinct from the four quadrature points on S^1. Thus, different values of the coefficients in the differential equation enter into the formation of the equations (5).

First, let $\mathcal{M}_1$ be the span of the four bilinear basis functions associated with the vertices of S^1 and $\mathcal{M}_2$ the span of the nine bilinear basis functions associated with the vertices of S^2_j, $j = 1, \ldots, 4$. Let us slightly generalize the question as to whether there exist $\mathcal{R}_{k+1}$ and $\mathcal{P}_k$ such that $A_k = \mathcal{R}_{k+1} A_{k+1} \mathcal{P}_k$ by asking if there exist maps

$$P : \mathcal{M}_1 \longrightarrow \mathcal{M}_2 \qquad \text{and} \qquad Q : \mathcal{M}_1 \longrightarrow \mathcal{M}_2$$

such that

$$(A_1 u, v) = (A_2 P u, Q v), \qquad u, v \in \mathcal{M}_1. \tag{7}$$

Consider a simple parameter count. Each of the matrices P and Q has 36 entries. For each nontrivial coefficient a_{ij}, b_i, or c, the quadrature rule associates sixteen values of the coefficient in the A_2-inner product and only four in the A_1-inner product. Thus, twelve independent constraints arise for each such coefficient. Since there are seven possibly nontrivial, distinct coefficients, it is clear that it cannot always be possible to satisfy (7). If there were fewer coefficients to handle, the maps could exist but have rather strange relationships to standard interpolation procedures.

Consider a different question. Let us take reasonable definitions of P and Q and ask to what extent (7) fails for locally smooth coefficients. Let $P = Q$ be the embedding operator between $\mathcal{M}_1$ and $\mathcal{M}_2$, and consider the special case for which

$$a_{ij}(x) = \delta_{ij} a(x), \qquad b_i(x) = c(x) = 0.$$

It follows easily from the Bramble-Hilbert lemma that

$$(A_1 u, v) - (A_2 P u, Q v) = \mathcal{O}(\|u\| \|v\|), \qquad u, v \in \mathcal{M}_1,$$

where the norm is the norm in H^1. If the analogous restriction and prolongation operators are used at each level,

$$\sigma_k^{(i)} = 1 + \mathcal{O}(h_k^2)$$

if the H^1-norm is employed at each level. Thus, using the naturally associated quadrature rule at each level is a reasonable choice for these choices for $\mathcal{R}_{k+1}$ and $\mathcal{P}_k$.

CONCLUSIONS

The theory here is more precise than in [1]. Further, it is applicable to problems that are not nested and/or ones in which the linear systems use quadrature in their assembly. Also, the theory here allows multigrid software (e.g., [6]) adaptively to change levels with a higher degree of precision than with the earlier theory.

The theory has been tested on several problems, ranging from simple (Poisson's equation on a rectangle) to quite difficult (a turbulent flame simulation). In each case, the theory has been very close to sharp in predicting what happens to the residual norm on the next finer level. Hence, we can conclude that this theory is useful in real computing situations in which level changes occur adaptively and standard theoretical models do not apply.

REFERENCES

[1] Douglas, C. C. and Douglas, J., A unified convergence theory for abstract multigrid or multilevel algorithms, serial and parallel, *SIAM J. Numer. Anal.*, 30:136–158, 1993.

[2] Bank, R. E. and Douglas, C. C., Sharp estimates for multigrid rates of convergence with general smoothing and acceleration, *SIAM J. Numer. Anal.*, 22:617–633, 1985.

[3] Douglas, C. C., Multi–grid algorithms with applications to elliptic boundary–value problems, *SIAM J. Numer. Anal.*, 21:236–254, 1984.

[4] Southwell, R. V., *Relaxation Methods in Engineering Science*, Oxford University Press, Oxford, 1940.

[5] Douglas, C. C., Douglas, J., and Fyfe, D. E., A multigrid unified theory for non-nested grids and/or quadrature, *E. W. J. Numer. Math.*, 2:285–294, 1994.

[6] Douglas, C. C., Implementing abstract multigrid or multilevel methods, in Melson, N. D., Manteuffel, T. A., and McCormick, S. F., editors, *Sixth Copper Mountain Conference on Multigrid Methods*, volume CP 3224, pp. 127–141, Hampton, VA, 1993, NASA.

THE EFFECTS OF DISSIPATION AND COARSE GRID RESOLUTION FOR MULTIGRID IN FLOW PROBLEMS

Peter Eliasson
FFA, Aeronautical Research Institute
Bromma, Sweden

Björn Engquist*
UCLA and Royal Institute of Technology
Stockholm, Sweden

SUMMARY

The objective of this paper is to investigate the effects of the numerical dissipation and the resolution of the solution on coarser grids for multigrid with the Euler equation approximations. The convergence is accomplished by multi-stage explicit time-stepping to steady state accelerated by FAS multigrid.

A theoretical investigation is carried out for linear hyperbolic equations in one and two dimensions. The spectra reveals that for stability and hence robustness of spatial discretizations with a small amount of numerical dissipation the grid transfer operators have to be accurate enough and the smoother of low temporal accuracy.

Numerical results give grid independent convergence in one dimension. For two-dimensional problems with a small amount of numerical dissipation, however, only a few grid levels contribute to an increased speed of convergence. This is explained by the small numerical dissipation leading to dispersion. Increasing the mesh density and hence making the problem over resolved increases the number of mesh levels contributing to an increased speed of convergence. If the steady state equations are elliptic, all grid levels contribute to the convergence regardless of the mesh density.

* Research sponsored by ARPA/ONR URI grant N00014-92-J-1890 and NSF DMS94-04942

INTRODUCTION

Multigrid methods have for a number of years been used to accelerate the convergence of the numerical solution to flow problems. This technique has been successfully applied to both subsonic and transonic speeds [6], [9]; however, in the hypersonic regime multigrid is sometimes less robust [12].

The objective in this paper is to investigate the convergence properties of primarily the Euler equations. The effects of the numerical dissipation and the resolution of the solution on coarser grids are two areas of concern, see e.g. [4]. It is also intended to investigate if multigrid in practice can give grid independent convergence [7] or, if not, how many grid levels contribute to an increased speed of convergence. The influence on the robustness and stability of the grid transfer operators (the restriction and prolongation) and the smoother are also addressed and investigated.

To analyze the solution of a hyperbolic system of equations linear scalar equations are studied. Central and upwind spatial discretizations are considered and the equations are integrated in time by an explicit multistage Runge-Kutta scheme which serves as a smoother. The damping properties are investigated for a number of different discretizations in space and time and for different restrictions and prolongations.

Numerical experiments are performed for two linear sets of equations in two dimensions that are hyperbolic in time. The steady state equations are hyperbolic and elliptic. Numerical results are also presented for a transonic and a hypersonic case solving the Euler equations. This paper is part of a doctoral thesis [3].

THE MULTIGRID METHOD

The FAS Multigrid Method

In the multigrid method several coarser grids are introduced by eliminating every other point on a finer grid. Assume that L grids are used. Each level in the multigrid method with a given grid is called a grid level. Denote the current grid level by l when $(1 \leq l \leq L)$, where $l = L$ is the finest grid and $l = 1$ is the coarsest grid. Let the finest grid L consist of N cells and hence the coarsest grid of $N/2^{L-1}$ cells. The FAS (Full Approximating Storage) multigrid algorithm by Brandt [1] for solving the problem

$$L_l v_l = f_l \tag{1}$$

can then be formulated as in [5]:

$$
\begin{array}{lll}
\textbf{procedure} FAS(l, v, f); & & \\
\textbf{if} \quad (l = 1) \textbf{ then} & & \\
\quad v := S(v, f, \nu_3) & & \textit{Smoothing on coarsest} \\
\textbf{else} & & \\
\quad v := S(v, f, \nu_1) & & Pre - smoothing \\
\quad w := r_l^{l-1} * v & & Restriction \\
\quad d := L_{l-1}(w) - r_l^{l-1} * (L_l(v) - f) & & Defect \\
\quad \bar{w} := w & & Initial\ guess \\
\quad \textbf{for } i := 1(1)\gamma \textbf{ do } FAS(l-1, \bar{w}, d) & & Recursive\ call \\
\quad v := v + p_{l-1}^l * (\bar{w} - w) & & Coarse\ grid\ correction \\
\quad v := S(v, f, \nu_2) & & Post - smoothing \\
\textbf{end}; & &
\end{array}
\tag{2}
$$

where S is the Runge-Kutta smoother, r_l^{l-1} is the restriction from the finer grid level l to the coarser level $l-1$, and p_{l-1}^l is the prolongation from level $l-1$ to l. A sawtooth cycle is considered with one pre-smoothing and no post-smoothing in the analysis, i.e. $\gamma = 1$, $\nu_1 = 1$, $\nu_2 = 0$, $\nu_3 = 1$. The algorithm (2) results in an iteration matrix M_l defined as

$$
\begin{array}{lll}
\text{Level } l > 1 & : & M_l = (I - p_{l-1}^l (I - M_{l-1}) L_{l-1}^{-1} r_l^{l-1} L_l) S_l \\
\text{Level } 1 & : & M_1 = S_1
\end{array}
\tag{3}
$$

The Grid Transfer Operators

Central operators for the prolongation and restriction are considered with the unknowns in the cell centers. The prolongation is denoted

$$a_l = p_{l-1}^l b_{l-1} \tag{4}$$

and the simplest prolongation in one dimension is the piecewise constant injection illustrated in Figure 1 where

$$a_{2j-1} = b_j, \quad a_{2j} = b_j \tag{5}$$

which is of order $m_p = 1$, i.e. it interpolates a polynomial of degree $m_p - 1$ exactly. We consider also a more accurate prolongation of degree $m_p = 2$ that interpolates a linear equation exactly

$$a_{2j} = \frac{1}{4}(3b_j + b_{j+1}), \quad a_{2j-1} = \frac{1}{4}(3b_j + b_{j-1}) \tag{6}$$

For the restriction operators the transpose of the prolongation operators are used

$$r_l^{l-1} = \frac{1}{2}(p_{l-1}^l)^T \tag{7}$$

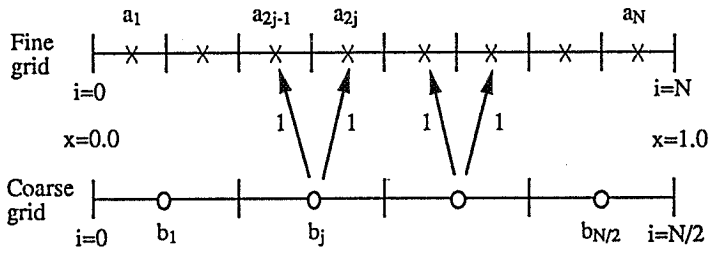


Figure 1: Prolongation from fine to coarse grid.

resulting in

$$b_j = \begin{cases} \frac{1}{2}(a_{2j-1} + a_{2j}) & , \quad m_r = 1 \\ \frac{1}{8}(a_{2j-2} + 3a_{2j-1} + 3a_{2j} + a_{2j+1}) & , \quad m_r = 2 \end{cases} \tag{8}$$

In [5] it is stated that the following condition must be fulfilled:

$$m_r + m_p > 2m \tag{9}$$

where $2m$ is the order of the differential operator. For convection problems, the Euler equations, and the model equations considered here, $2m = 1$, i.e. the piecewise constant prolongation ($m_p = 1$) and the restriction ($m_r = 1$) can in theory be used. It will turn out, though, that interpolation of higher degree of accuracy is stabilizing.

The Smoother

Explicit Runge-Kutta time stepping is used as a smoother in the multigrid cycle to accelerate the convergence to steady state. To solve for the steady state equation $L(v) = 0$, v^n is iterated in time using an m-stage Runge-Kutta scheme defined as

$$\begin{aligned} v^{(0)} &= v^n \\ v^{(1)} &= v^{(0)} - \alpha_1 \Delta t L(v^{(0)}) \\ &\vdots \\ v^{(m)} &= v^{(0)} - \alpha_m \Delta t L(v^{(m-1)}) \\ v^{n+1} &= v^{(m)} \end{aligned} \tag{10}$$

where the coefficients α_i, $i = [1, 2, \ldots, m]$, are chosen to make the smoothing efficient.

DAMPING PROPERTIES IN 1D

A Scalar 1D Test Problem

To analyze the behavior of a hyperbolic system of conservation laws a simpler scalar one-dimensional test equation is considered:

$$\begin{aligned} \frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} &= 0; \quad t \geq 0, \ 0 \leq x \leq 2\pi \\ u(x,0) &= u_0(x) \end{aligned} \tag{11}$$

with periodic boundary conditions for the Fourier analysis below.

The semidiscrete form of (11) can be written as

$$\frac{d}{dt}v_j + \frac{1}{h}(f^*_{j+\frac{1}{2}} - f^*_{j-\frac{1}{2}}) = 0 \tag{12}$$

where

$$f^*_{j+\frac{1}{2}} = \frac{1}{2}(v_j + v_{j+1}) - \frac{1}{2}Q\Delta v_{j+\frac{1}{2}} + \kappa^{(4)}\Delta^3 v_{j+\frac{1}{2}}$$

represents a cell face flux. Δ is a central difference operator. $Q = 1$ results in a first order accurate upwind scheme; $\kappa^{(4)} = 0$ for that scheme usually. Second order accurate upwind schemes can be obtained by using limiters resulting in a non-linear scheme. For the analysis, Q is considered to be a constant in order to have a linear scheme. A central difference scheme with artificial dissipation is obtained when $Q = 0$ and $\kappa^{(4)}$ is a small positive constant.

Damping of Smooth Waves

To study the damping properties of the multigrid cycle in (2), a Fourier transform of the iteration matrix M_l is considered denoted $\tilde{M}_l$. By coupling frequencies pairwise between a finer and a coarser grid the transformed iteration matrix $\tilde{M}_l$ becomes a block diagonal matrix with $2^{l-1} \times 2^{l-1}$ matrices on the diagonal. The damping properties are investigated by calculating the eigenvalues to $\tilde{M}_l$.

The high frequency errors are damped by the smoother, the Runge-Kutta scheme. The low frequencies (or the smooth waves) are not damped very well, but on the other hand it has been shown [8] that the smooth waves increase their speed using multigrid by a factor of $2^l - 1$ for the sawtooth cycle. Under some general conditions [3] it is possible to derive an expression for the largest eigenvalue to the transformed iteration matrix $\tilde{M}_l$ for smooth waves :

$$\max_j |\lambda_j| = 1 - (\sigma\xi_l)^2 \left(C_l(\beta_2 - \frac{1}{2}) + A_l \frac{Q_l}{2\sigma} \right) + O(\xi^3) \tag{13}$$

provided that the frequency $\xi_l = \omega h_l$ is small enough, where ω is the wave number and h_l is the constant cell length on the finest level l. σ is the CFL number, β_2 is the constant for the square term in the Runge-Kutta polynomial $p(z) = 1 + z + \sum_{i=2}^{i=m} \beta_i z^i$

to an m-stage Runge-Kutta scheme. For a consistent Runge-Kutta scheme $\beta_2 = \alpha_{m-1}$ in (10). Q_l is the constant Q in (11) on grid level l. C_l and A_l are two constants:

$$A_l = 2^l - 1 \quad , \quad C_l = \frac{4^l - 1}{3} \tag{14}$$

that grow exponentially with the number of grid levels.

It is clear from (14) that β_2 should be chosen $\beta_2 > 0.5$ for good damping which is the same as requiring the Runge-Kutta scheme to be first order accurate. It is also clear that the multigrid increases the damping due to the factors A_l and C_l in (13). As could be expected most damping is obtained from a first order accurate upwind scheme where $Q_l = 1$. It can also be seen that the numerical dissipation on coarser grids does not contribute to the damping of the smoothest waves. Only the term Q_l on the finest grid appears in (13). This is true also for other values of ν_1, ν_2, ν_3 in (2). Consequently, it is not possible to increase the damping of the lowest frequencies by using a scheme with more numerical dissipation on coarser grids.

Even though the damping of smooth waves is increased by multigrid the propagation of smooth waves dominates over the damping, which is illustrated in Figure 2. A smooth wave is transported 90 iterations using one grid and 6 iterations with four grids using an upwind discretization $Q = 1, \kappa^{(4)} = 0$. The step length is $h = \frac{1}{256}$. Since the speed of the smooth wave is increased by a factor of 15 the wave is transported the same distance in the one-grid and four-grid cases. A three stage Runge-Kutta method ($\beta_2 = 0.6$) is used and $CFL = \sigma = 1$.

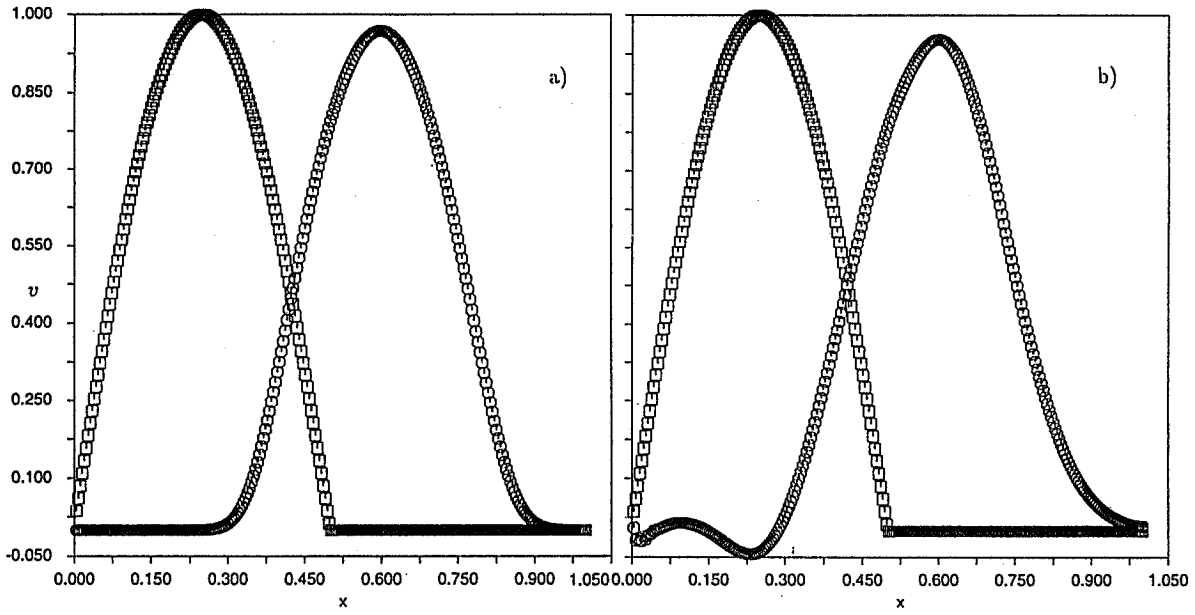


Figure 2: Propagation of a smooth wave to the right with the upwind scheme, $Q = 1$, a) 1 grid and 90 iterations, b) 4 grids and 6 iterations.

Eigenvalues to the Iteration Matrix

To illustrate the damping properties for different spatial discretizations and grid transfer operators the absolute value of the eigenvalues to the Fourier transformed iteration matrix of the sawtooth cycle are plotted. For all cases a 5-stage Runge-Kutta scheme with the coefficients $(0.0814, 0.191, 0.342, 0.574, 1.)$ is used [3] which is first order accurate and provides optimal high frequency damping for both an upwind scheme ($Q = 1$) and a central scheme ($\kappa^{(4)} = \frac{1}{16}$) for a CFL just above 2. The region of stability for this scheme is shown in Figure 3. The scheme has the advantage that different spatial discretizations can be used on different grid levels. In Figure 4

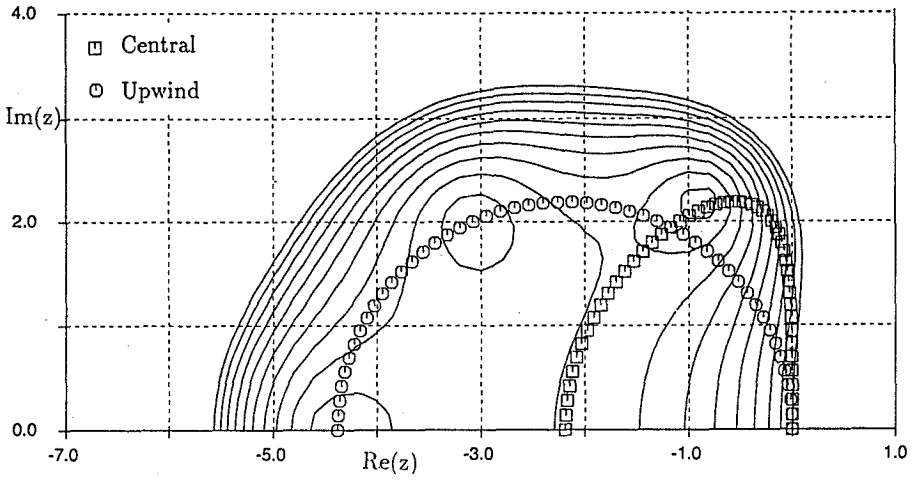


Figure 3: Region of stability and locus of differencing operator for an upwind ($Q = 1$, $\kappa^{(4)} = 0$) and a central ($Q = 0$, $\kappa^{(4)} = \frac{1}{16}$) scheme with five stages, $\Delta|p(z)| = 0.1$.

the damping of a central scheme is shown using a piecewise constant prolongation ($m_p = 1$) and its transpose for restriction ($m_r = 1$) with $CFL = 2$. The absolute values of the eigenvalues to the matrices $\tilde{M}_1$, $\tilde{M}_2$, $\tilde{M}_3$, and $\tilde{M}_4$ are shown as a function of the frequency where $\tilde{M}_1$, the single grid case, is represented with a solid line in all figures. The increased damping of the lowest frequencies can clearly be seen.

Even though the central scheme in Figure 4 is stable for grid levels $1 - 4$ with $CFL = 2$, divergence is obtained for lower CFL numbers. Figure 5 shows the same scheme but with $CFL = 1.25$. This scheme is obviously unstable using 3 and 4 grids. By increasing the accuracy of the prolongation and/or the restriction the scheme is stabilized. The central scheme with $\kappa^{(4)} = \frac{1}{16}$ is a scheme with a rather small amount of dissipation. One could also use the dissipative first order upwind scheme on the coarser grids to stabilize.

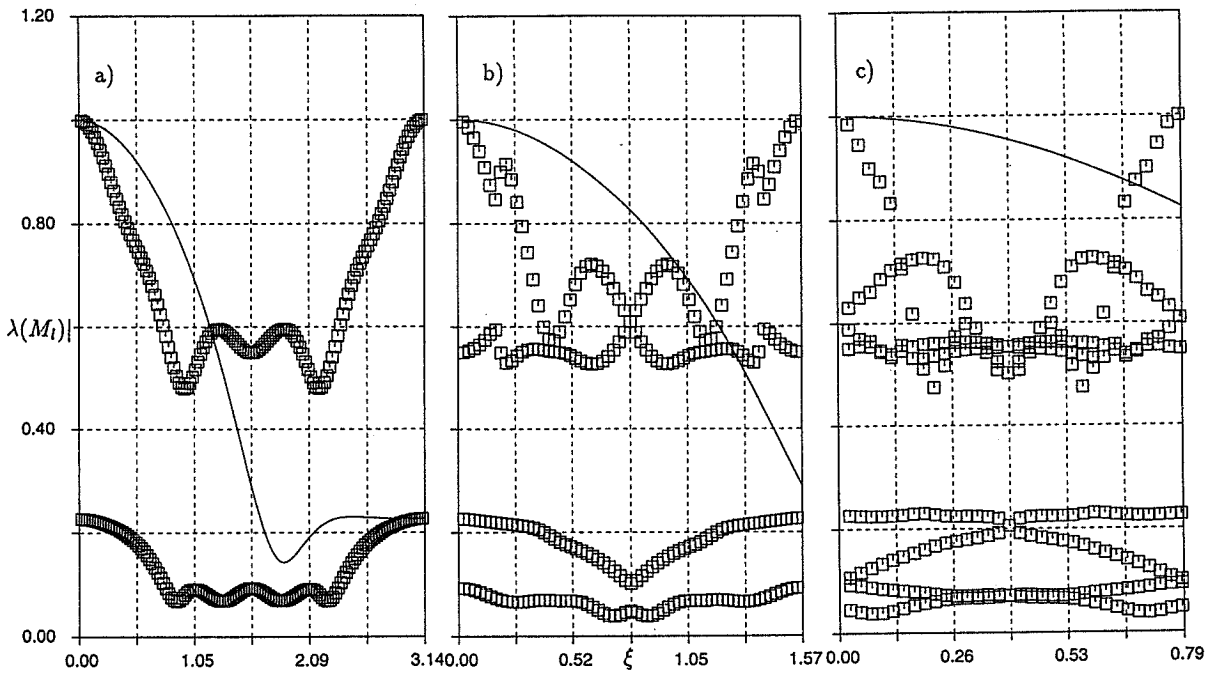


Figure 4: Eigenvalues of a) $\tilde{M}_2$, b) $\tilde{M}_3$, c) $\tilde{M}_4$. $\tilde{M}_1$ is the solid line in all three plots. Central scheme is used, $\kappa^{(4)} = \frac{1}{16}$, $Q = 0$, $CFL = 2$, $m_p = 1$, $m_r = 1$.

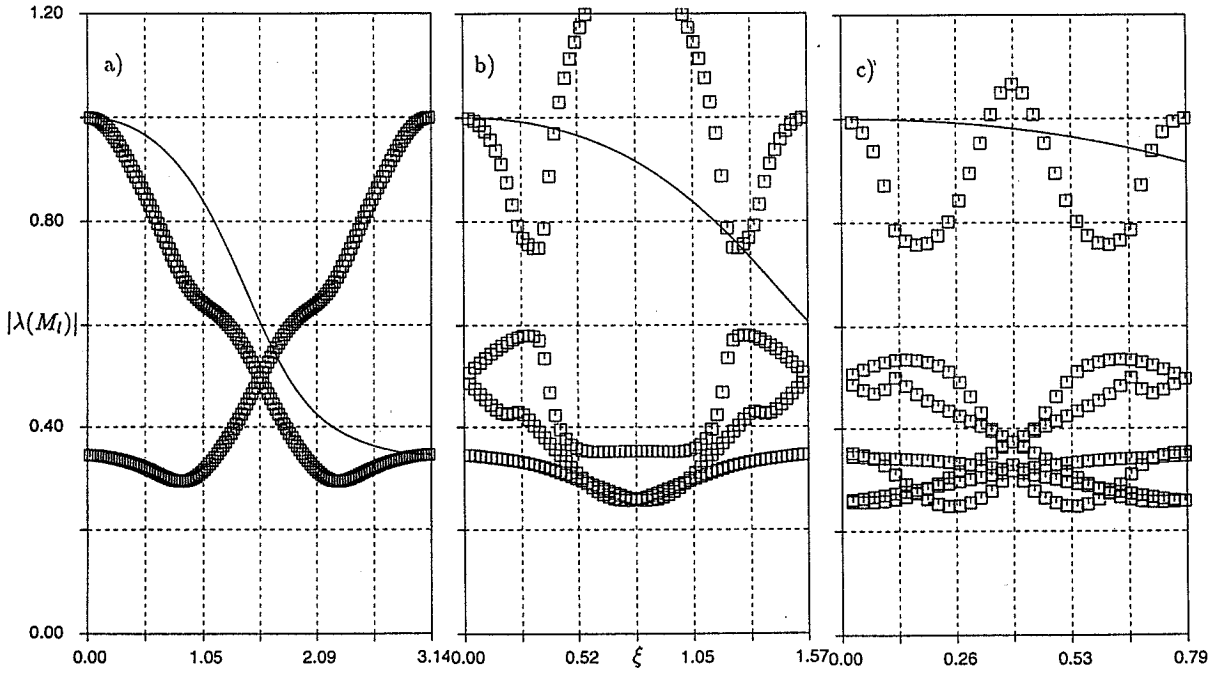


Figure 5: Eigenvalues of a) $\tilde{M}_2$, b) $\tilde{M}_3$, c) $\tilde{M}_4$. $\tilde{M}_1$ is the solid line. Central scheme is used, $\kappa^{(4)} = \frac{1}{16}$, $Q = 0$, $CFL = 1.25$, $m_p = 1$, $m_r = 1$.

The spectra for the central scheme in Figure 5 are plotted in Figure 6 with different levels of accuracy for the prolongation and restriction. The eigenvalues outside the region of stability can clearly be seen for the low order accurate grid transfer operators; the scheme is stabilized and the spectra brought closer to the single grid spectra as more accurate grid transfer operators are used. Figure 7 shows the spectra for an upwind method where the other conditions are the same as in Figure 6. This dissipative scheme is stable for all prolongations, restrictions, and single grid stable CFL numbers.

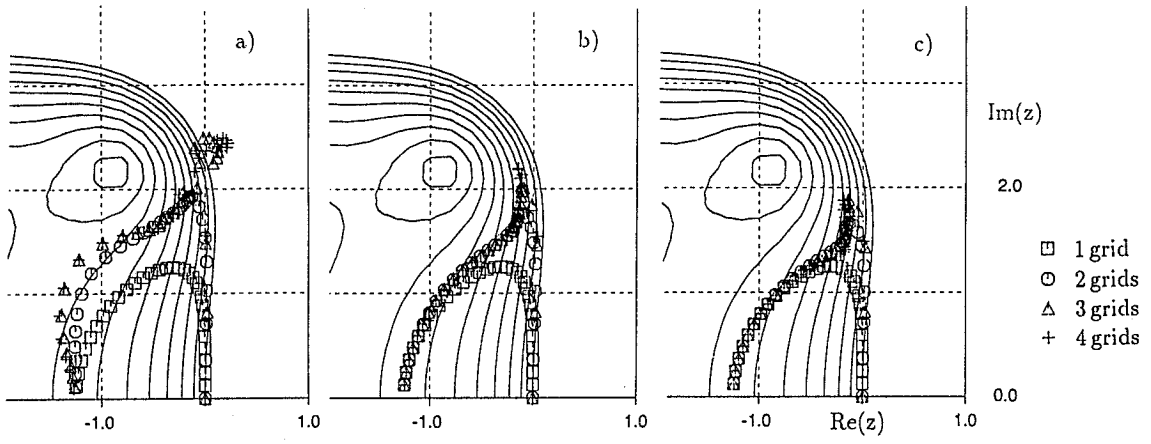


Figure 6: Spectra for single-grid, two-grid, three-grid and four-grid multigrid. $\kappa^{(4)} = \frac{1}{16}$, $Q = 0$, $CFL = 1.25$. a) $m_p = 1$, $m_r = 1$, b) $m_p = 2$, $m_r = 1$, c) $m_p = 2$, $m_r = 2$.

The following can be established. A certain amount of dissipation has to be introduced to the system. A large amount of numerical dissipation can be used, e.g., by choosing a first order accurate upwind scheme, which stabilizes the multigrid iterations. If a good converged solution is desired, however, the amount of numerical dissipation on the finest grid has to be rather small to avoid smearing the solution. The multigrid cycle is then stabilized by increasing the accuracy of the grid transfer operators, by choosing a Runge-Kutta scheme with a low order of accuracy and good high frequency damping, and by doing several Runge-Kutta sweeps on coarser grids.

DAMPING PROPERTIES IN 2D

A Scalar 2D Test Problem

The 1D hyperbolic scalar equation in (11) is extended to 2D as:

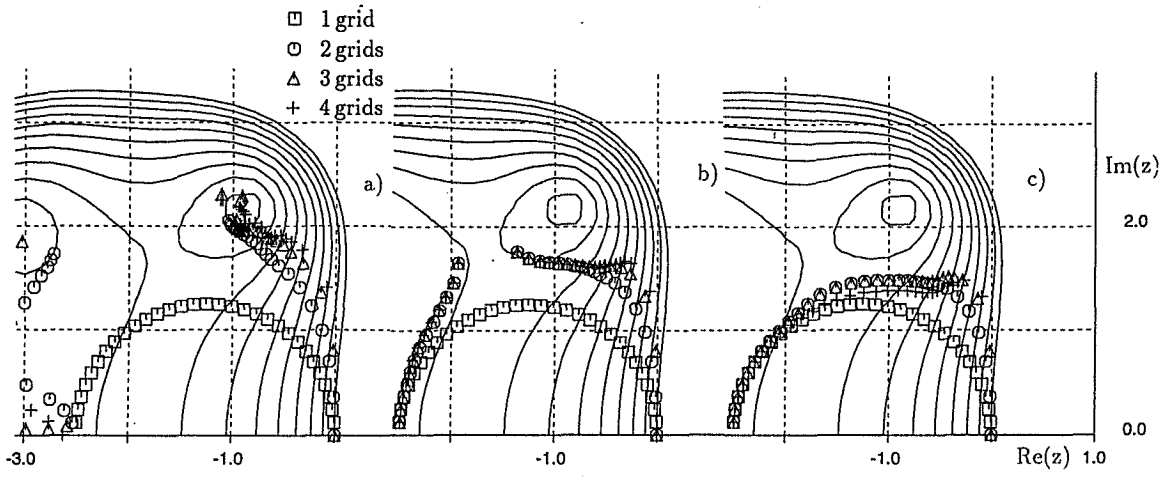


Figure 7: Spectra for single-grid, two-grid, three-grid and four-grid multigrid. $\kappa^{(4)} = 0$, $Q = 1$, $CFL = 1.25$. a) $m_p = 1$, $m_r = 1$, b) $m_p = 2$, $m_r = 1$, c) $m_p = 2$, $m_r = 2$.

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} + a\frac{\partial u}{\partial y} = 0, \quad t \geq 0, \quad 0 \leq x, y \leq 2\pi \tag{15}$$

where a is some constant. The discretization of (15) follows the 1D counterpart; a similar Fourier analysis can be made assuming periodic boundary conditions with the prolongation and restriction extended in a straightforward manner to 2D.

Damping of Smooth Waves

Some important observations can be made from a Taylor expansion of small frequencies of the largest eigenvalue to the iteration matrix. In 2D it is possible for the Fourier transformed exact operator $i(\xi_x + a\xi_y)$ to vanish or to become very small. ξ_x, ξ_y are the frequencies in x, y. For a two-grid problem where the problem is solved exactly on the coarse grid and the exact operator vanishes this implies that the two-grid algorithm can only reduce the residual by a factor of $\frac{1}{2}$ for the first order upwind scheme ($Q = 1$), even though the problem is solved exactly on the coarse grid. The situation is even worse for the central scheme which is only reduced by a factor of $\frac{7}{8}$. This is fundamentally different than in 1D where the residual is reduced $O(\xi)$.

This observation has also been made by Decker & Turkel [2]. They point out that a fourth difference dissipation on the fine grid and a second difference dissipation on the coarser grid can lead to a situation with practically no damping. This means that

the convergence rate per multigrid cycle cannot be made arbitrarily small. Since the damping cannot be made arbitrarily small for a two-grid method with an exact solution on the coarsest grid, there is consequently no use in making too many smoothing sweeps in 2D multigrid on coarser grids.

Similar observations have been made by Mulder [10], [11]. Mulder notices that when the exact operator vanishes the discretization is of the order of the truncation error. The order of accuracy of the restriction and/or prolongation must then be increased. The equation (9) is no longer valid since one is actually looking at the truncation error which can be viewed as a discretization of a higher order differential equation with a value $2m > 1$ for these waves. Mulder shows that the worst-rate convergence can be estimated to $1 - 2^{-p}$ where p is the spatial order of accuracy. This agrees with what is found above; for a first order scheme ($p = 1$) this is $\frac{1}{2}$. When the exact operator vanishes and the remaining operator is the fourth difference operator a third order accurate scheme is obtained which corresponds to $1 - 2^{-3} = \frac{7}{8}$.

Both Decker & Turkel and Mulder conclude that the above estimates are too pessimistic since they are worst case estimates. In real applications with non-periodic boundary conditions, better rates of convergence are usually obtained.

NUMERICAL EXPERIMENTS

The 2D equation (15) is used for numerical experiments:

$$\begin{aligned} &\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} + a\frac{\partial u}{\partial y} = 0; && t \geq 0, \;\; 0 \leq x, y \leq 1 \\ &u(x, y, 0) = 0 \\ &u(0, y, t) = u_0(y) = \sin(2\pi m y) \\ &u(x, 0, t) = u(x, 1, t) \end{aligned} \tag{16}$$

where $a \geq 0$ and $m \geq 0$ is an integer that determines the number of wave lengths along the y-axis. The sine wave along the y-axis will propagate into the domain along characteristics in a direction that depends on a. The exact steady state solution to (16) is $\bar{u} = u_0(y - ax)$. The convergence of the 1D equation (11) gives almost grid independent results as the grid is refined [3] and is therefore omitted here.

The idea with this test case was to see how the convergence is influenced as the number of grids increases. More specifically, what happens when the sine wave in (16) is poorly or not at all resolved on coarser grids? How many grids can be used and how does the dissipation influence the rate of convergence?

Figure 8 shows the converged solutions on the different grids at $y = 0$ for a central scheme where $a = 0.5$, $m = 4$, $\kappa^{(4)} = \frac{1}{16}$, and $Q = 0$. The converged solution is well represented on the three finest grids; the deviation from the exact solution then grows as the size of the grid is reduced.

In Figure 9 the rate of convergence is plotted for the same central scheme for grid levels $1 - 5$. A linear prolongation ($m_p = 2$) is used with a lower order restriction

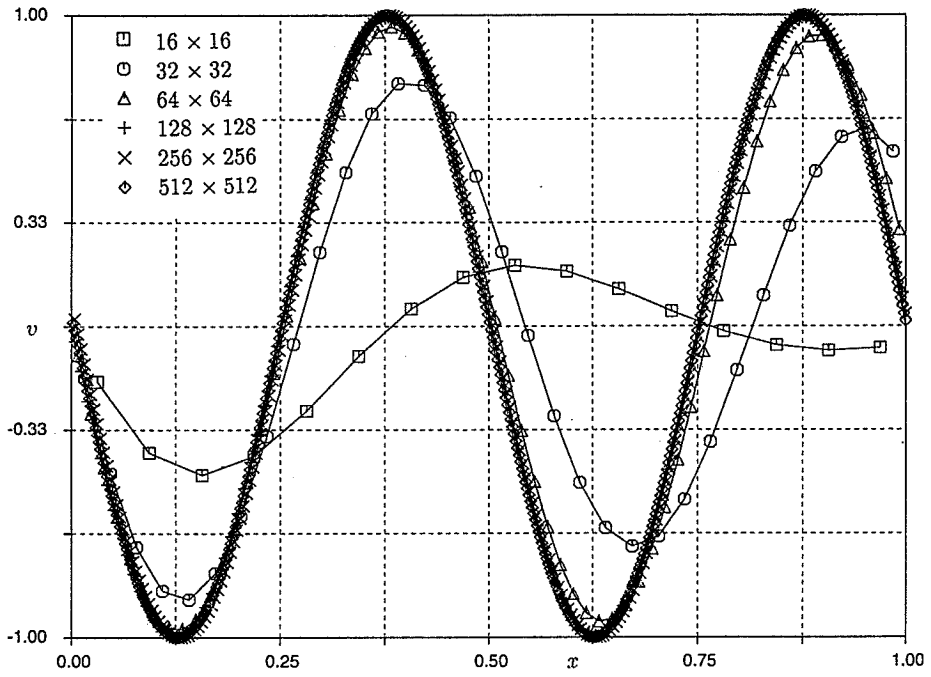


Figure 8: Converged solution for the 2D scalar equation at $y = 0$ for the central scheme $\kappa^{(4)} = \frac{1}{16}$, $Q = 0$. Constants $a = 0.5$, $m = 4$.

($m_r = 1$). This case does not converge if $m_p = 1$ even though the Fourier analysis in Figure 4 gives stable eigenvalues. The CFL number is $CFL = 2$ using the five-stage scheme previously mentioned; sawtooth cycles are used. The convergence rates are shown as the fine grid size is increased from $N = 64$ to $N = 512$ cells in each direction. The convergence is very slow until the low frequency errors are pushed out of the computational domain. ¿From then on the high frequency errors that are left in the domain are effectively damped by the smoother leading to a fast convergence. For the finest grid the best convergence rate is achieved using three grid levels. However, kinks on the curves of convergence for the four-grid and five-grid cases occur after a while, which make these cases require more iterations than the three-grid case. Notice also that multigrid gives almost no speedup for the coarser grid.

Further increase in the accuracy of the grid transfer operators has a very small influence on the convergence rate. However, if the dissipation is increased on the finest grid as shown in Figure 10, the kinks become smaller and almost vanish for the first order upwind scheme where all grid levels contribute to an increased speed of convergence. The solution for this dissipative scheme is poorly. resolved, though, on all grids [3].

If the residuals are to be brought to machine accuracy, the amount of dissipation is very important to gain from several grid levels in multigrid. If, on the other hand, the iterations are to be interrupted when the error reaches the level of truncation error, then almost all grid levels will contribute to the convergence [3].

For a hyperbolic linear system of equations where the steady state equations are elliptic the situation is different. Equations (17) are hyperbolic in time but converge

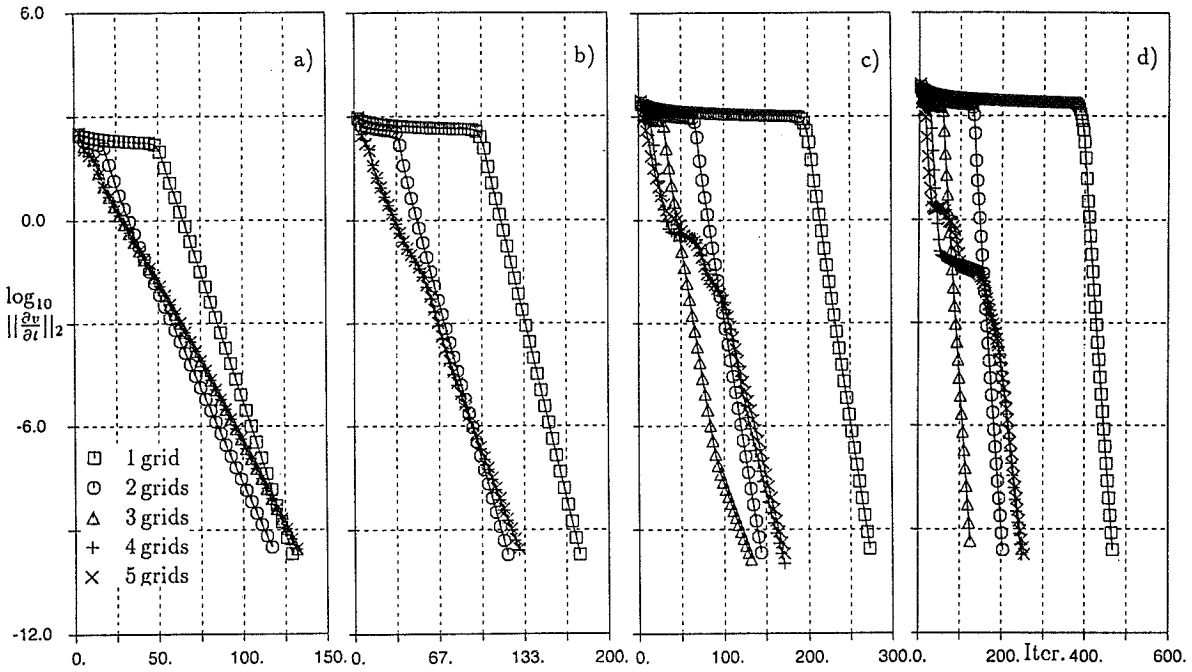


Figure 9: Rate of convergence with the central scheme, $\kappa^{(4)} = \frac{1}{16}$, $Q = 0$, $a = 0.5$, $m = 4$, $CFL = 2.0$. $m_p = 2$, $m_r = 1$. a) $N = 64$. b) $N = 128$. c) $N = 256$. d) $N = 512$.

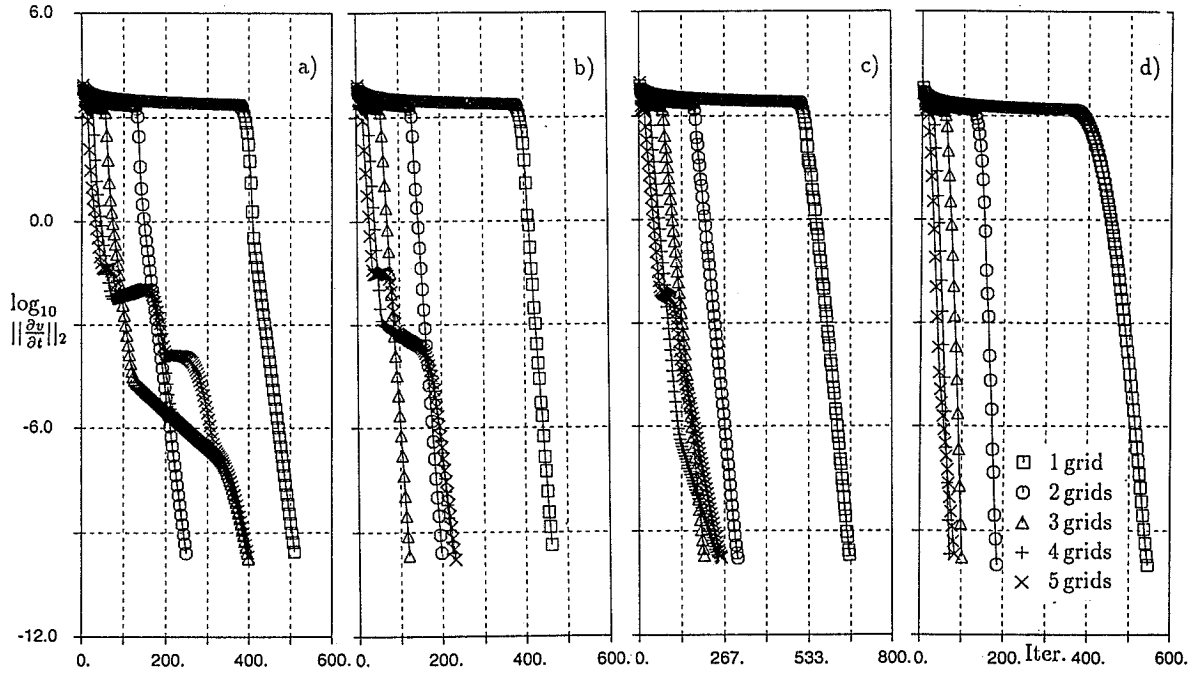


Figure 10: Rate of convergence, $a = .0.5$, $m = 1$, $CFL = 2.0$, $m_p = 2$, $m_r = 2$, $N = 512$. a) Central, $\kappa^{(4)} = 0.01$. b) Central, $\kappa^{(4)} = 0.0625$. c) Central, $\kappa^{(4)} = 0.2$. $(CFL = 1.5)$. d) Upwind, $Q = 1$.

to the elliptic Laplace equation. The equations are solved in the same way as the 2D scalar. As can be seen from Figure 11 there is a speedup from all grid levels.

$$\frac{\partial}{\partial t}\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} + \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}\frac{\partial}{\partial x}\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} + \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\frac{\partial}{\partial y}\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = 0, \quad t \geq 0, \;\; 0 \leq x, y \leq 1$$
$$u_1(x,y,0) = u_2(x,y,0) = 0$$
$$u_1(0,y,t) = \sin(2\pi m y)$$
$$u_2(1,y,t) = 0$$
$$u_1(x,0,t) = u_1(x,1,t)$$
$$u_2(x,0,t) = u_2(x,1,t) \tag{17}$$

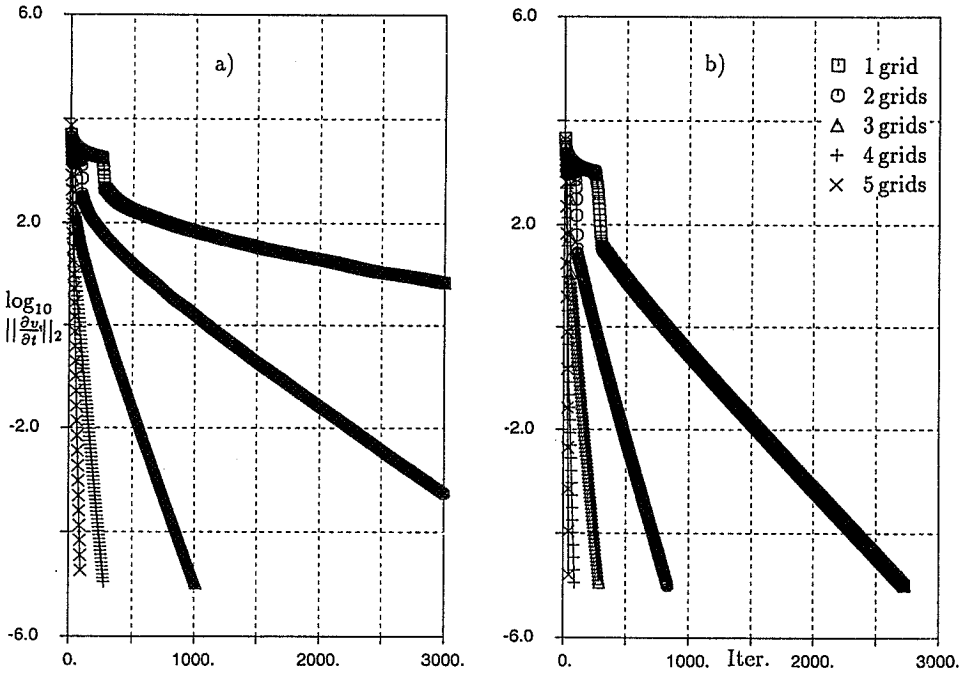


Figure 11: Rate of convergence with 2D elliptic equation, $m = 4$. Five stage Runge-Kutta, $CFL = 2.0$, $N = 256$. a) central scheme $\kappa^{(4)} = \frac{1}{16}$, $Q = 0$, $m_p = 2$, $m_r = 1$. b) upwind scheme $\kappa^{(4)} = 0$, $Q_{j+\frac{1}{2}} = 1$, $Q_{k+\frac{1}{2}} = a$, $m_p = 1$, $m_r = 1$.

For all cases the resolution of the solution on coarser grids appears to be of small practical importance. The difference between the solutions on coarser and finer grids lies in the high frequencies damped by the smoother. If a large amount of numerical dissipation for the hyperbolic steady state problem is used, the convergence rates are similar to the ones of the elliptic steady state problem. The high frequencies are well damped and all grid levels contribute to an increased speed of convergence even though the solution is poorly resolved on coarser grids. If a smaller amount of numerical dissipation is used, however, the solution itself is better represented on coarser grids. There is still a small difference in the solutions on finer and coarser grids. The difference lies in the intermediate frequencies that are not damped very well by the smoother. This difference and the fact that the exact operator can vanish cause the convergence with multigrid to deteriorate for hyperbolic steady state problems.

Convergence close to being grid independent is only obtained in one dimension and for the elliptic steady state problem.

Finally some results for the Euler equations are presented. In Figure 12 the rate of convergence is plotted for a 2D transonic calculation over a NACA 0012 airfoil at a Mach number of 0.8 and an angle of attack $\alpha = 1.25°$. The steady state equations are mixed hyperbolic/elliptic in the dominating subsonic region, and the convergence resembles, to a large extent, the linear elliptic steady state case, above which all grid levels contribute to the convergence.

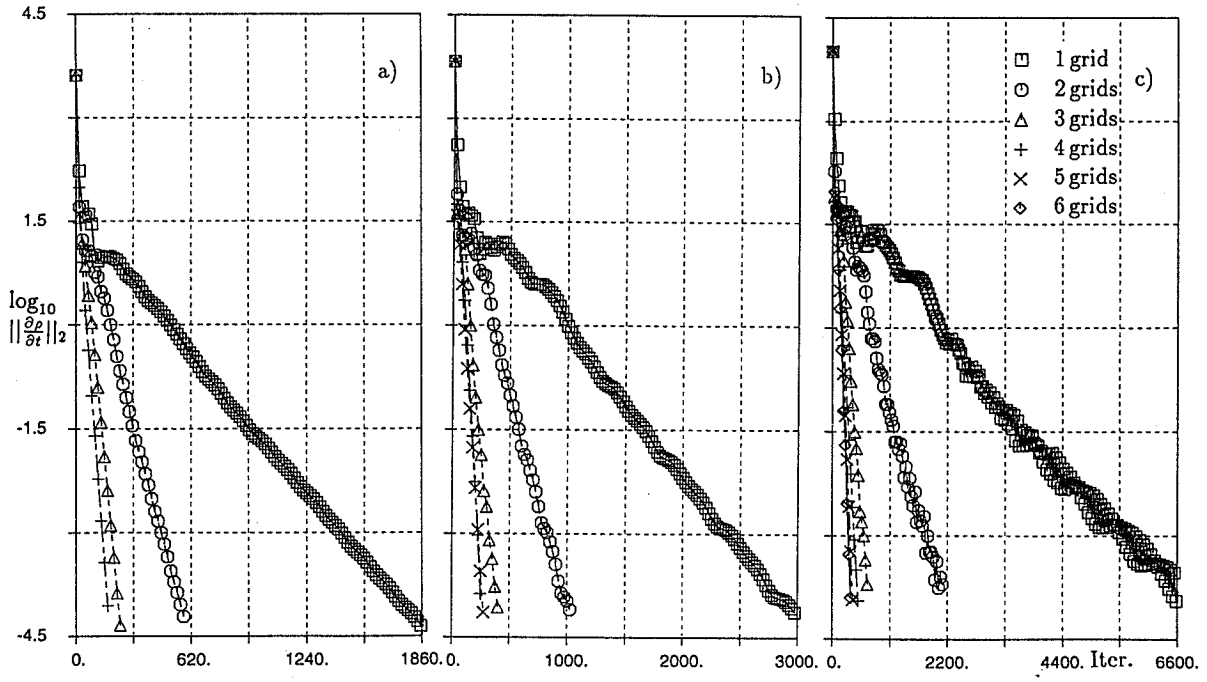


Figure 12: Rate of convergence over the NACA 0012 airfoil using central scheme, $\kappa^{(2)} = 0.25$, $\kappa^{(4)} = \frac{1}{16}$. Five stage Runge-Kutta with $CFL = 2.0$. $m_p = 1$, $m_r = 2$. a) Fine grid 65×17. b) Fine grid 129×33. c) Fine grid 257×65.

Figure 13 shows the rate of convergence for a hypersonic hyperboloid-flare problem at a Mach number of 8.7 [3]. This problem is axisymmetric and represents the nose of a space shuttle. The flow is supersonic almost everywhere. As can be seen the finest grid has to be fine enough to gain from multigrid in accordance with the hyperbolic steady state problem above.

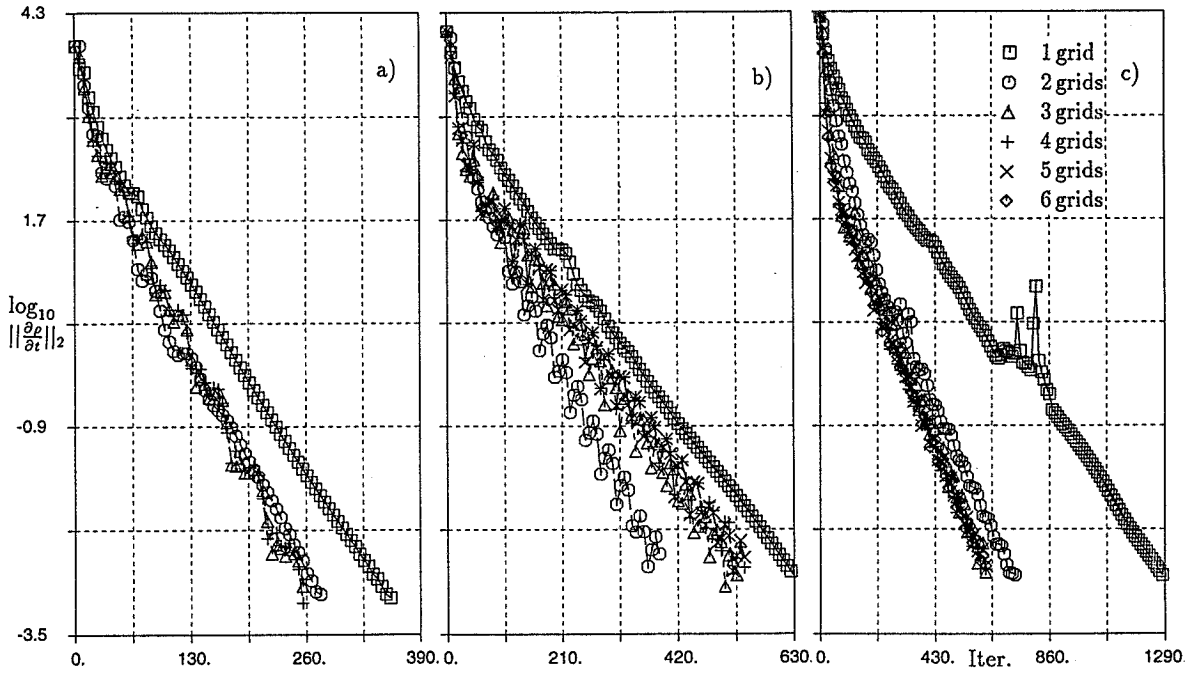


Figure 13: Rate of convergence for the hyperboloid flare. Five stage Runge-Kutta with $CFL = 1.5$. Central scheme with $\kappa^{(2)} = 1.0$, $\kappa^{(4)} = 0.0625$, $m_p = 2$, $m_r = 2$. a) finest grid 33×17. b) finest grid 65×33. a) finest grid 129×65.

CONCLUSION

The objective was to investigate the influence of the numerical dissipation and the resolution of the solution on coarser grids for flow problems with multigrid.

If a low amount of numerical dissipation is used on the fine grid the multigrid cycle is stabilized by increasing the accuracy of the grid transfer operators, by using a Runge-Kutta scheme with a low order of accuracy and/or by adding more numerical dissipation on coarser grids. For a higher amount of numerical dissipation, the multigrid cycle is stable in any case.

Numerical results for model problems give grid independent convergence only in one dimension and for the 2D elliptic steady state problem. For the 2D hyperbolic steady state problem with moderate numerical dissipation the convergence is grid independent down to the level of truncation error but deteriorates in multigrid when converged further. Only a few grid levels where the solution is over resolved contribute to an increased speed of convergence. This is explained by the small numerical dissipation leading to dispersion and a vanishing exact operator. The convergence behavior for the Euler equations was similar to that of the model problems.

REFERENCES

[1] Brandt, A. "Multi-Level Adaptive Solutions to Boundary Value Problems," *Math. Comp.*, Vol. 31, 1977, pp. 333-390.

[2] Decker, N. and Turkel, E. "Multigrid for Hypersonic Inviscid Flows," *Proceedings of 3rd International Conference on Hyperbolic Problems*, Uppsala, 1990.

[3] Eliasson, P. "Dissipation Mechanisms and Multigrid Solutions in a Multiblock Solver for Compressible Flow," *Doctoral Thesis*, TRITA-NA-R9314, ISSN-0348-2952, 1993.

[4] Gustafsson, B. and Lötstedt, P. "Analysis of the Multigrid Method Applied to First Order Systems," *Proc. of the Fourth Copper Mountain Conf. on Multigrid Methods*, Eds. J. Mandel *et al.*, SIAM, Philadelphia, 1989, pp. 181-233.

[5] Hackbush, W. "*Multi-Grid Methods and Applications*," Springer, 1989.

[6] Jameson, A. "Transonic Flow Calculations for Aircraft, Lecture Notes in Mathematics," *Numerical Methods in Fluid Dynamics*, Vol. 1127, 1985, pp. 156-242.

[7] Lötstedt, P. "Grid Independent Convergence of the Multigrid Method for First-Order Equations," *SIAM Journal of Numerical Analysis*, Vol. 29, No. 5, 1992, pp. 1370-1394.

[8] Lötstedt, P. and Gustafsson, B. "Fourier Analysis of Multigrid Methods for General System of PDE," *Mathematics of Computation*, Vol. 60, No. 202, 1993, pp. 473-493.

[9] Mulder, W. A. "Multigrid Relaxations for the Euler Equations," *Journal of Computational Physics*, Vol. 60, 1985, pp. 232-252.

[10] Mulder, W. A. "Multigrid, Alignment, and Eulers Equations," *Proc. of the Fourth Copper Mountain Conf. on Multigrid Methods*, Eds. J. Mandel *et al.*, SIAM, Philadelphia, 1989, pp. 348-364.

[11] Mulder, W. A. "A High-Resolution Euler Solver Based on Multigrid, Semi-Coarsening, and Defect-Correction," *Journal of Computational Physics*, Vol. 100, 1992, pp. 91-104.

[12] Radespiel, R. and Swanson, R. C. "Progress with Multigrid Schemes for Hypersonic Flow Problems," *ICASE Report 91-89*, 1991.

Page intentionally left blank

MULTIGRID AND KRYLOV SUBSPACE METHODS FOR THE DISCRETE STOKES EQUATIONS

HOWARD C. ELMAN*

Abstract. Discretization of the Stokes equations produces a symmetric indefinite system of linear equations. For stable discretizations, a variety of numerical methods have been proposed that have rates of convergence independent of the mesh size used in the discretization. In this paper, we compare the performance of four such methods: variants of the Uzawa, preconditioned conjugate gradient, preconditioned conjugate residual, and multigrid methods, for solving several two-dimensional model problems. The results indicate that where it is applicable, multigrid with smoothing based on incomplete factorization is more efficient than the other methods, but typically by no more than a factor of two. The conjugate residual method has the advantage of being both independent of iteration parameters and widely applicable.

Key words. Stokes, multigrid, Krylov subspace, conjugate gradient, conjugate residual, Uzawa

1. Introduction.

Consider the system of partial differential equations

$$\begin{aligned} -\Delta u + \nabla p &= f && \text{on } \Omega \\ -\operatorname{div} u &= 0 && \\ u &= 0 && \text{on } \partial\Omega, \\ \textstyle\int_\Omega p &= 0 && \end{aligned} \tag{1}$$

where Ω is a simply connected bounded domain in $\mathbf{R}^d$, $d = 2$ or 3. This system, the *Stokes equations*, is a fundamental problem arising in computational fluid dynamics; see, e.g., [7, 12, 14, 17]; u is the d-dimensional velocity vector defined on Ω, and p represents pressure.

Discretization of (1) by finite difference or finite element techniques leads to a linear system of equations of the form

$$\begin{pmatrix} A & B^T \\ B & -C \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix}, \tag{2}$$

where A is a set of uncoupled discrete Laplacian operators and C is a positive semidefinite matrix. We consider here only *stable* discretizations, i.e., those for which the condition number of the Schur complement matrix $BA^{-1}B^T + C$ is bounded independently of the mesh size used in the discretization. For finite element discretizations with $C = 0$, this is a consequence of the *inf-sup condition* and upper bound

$$\gamma \le \inf_q \sup_v \frac{(q, \operatorname{div} v)}{|v|_1 \, \|q\|_0}, \qquad \frac{|(q, \operatorname{div} v)|}{|v|_1 \, \|q\|_0} \le \Gamma,$$

where γ and Γ are independent of the mesh size. Here, $|\cdot|_1$ and $\|\cdot\|_0$ denote the H^1 seminorm and L_2 norm, respectively, on the discrete velocity and pressure spaces,

* Department of Computer Science and Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742, e-mail: elman@cs.umd.edu. This work was supported by the U. S. Army Research Office under grant DAAL-0392-G-0016, and by the National Science Foundation under grant ASC-8958544.

and the bounds are taken over all v and q in the appropriate discrete spaces; see [7, 12, 14, 17].

In recent years, a variety of iterative algorithms have been devised for solving the discrete Stokes equations. In this paper, we compare the performance of four such methods:

1. a variant of the Uzawa method;
2. a preconditioned conjugate gradient (PCG) method applied to a transformed version of (2);
3. a preconditioned conjugate residual (PCR) method;
4. multigrid (MG).

The Uzawa method is the first among these to have been devised [2] and it is often advocated as an efficient solution technique, see e.g. [7, 12, 14]. The convergence factor associated with it is proportional to $(\kappa - 1)/(\kappa + 1)$ where κ is the condition number of the Schur complement $BA^{-1}B^T + C$ (see §2.5). The conjugate gradient method, developed by Bramble and Pasciak [5], has a convergence factor proportional to $(\sqrt{\kappa} - 1)/(\sqrt{\kappa} + 1)$ but a larger cost per step than the Uzawa method. The preconditioned conjugate residual method was developed by Rusten and Winther [24], Silvester and Wathen [26], and Wathen and Silvester [31], and its convergence behavior is determined by properties of the indefinite matrix. For multigrid, we consider versions derived from two smoothing strategies: a variant of the distributed Gauss-Seidel method of Brandt and Dinar [6], and the technique based on incomplete factorization developed by Wittum [35]; we refer to these as MG/DGS and MG/ILU, respectively.

These methods all have the property that for appropriate choice of preconditioners (or for multigrid, smoothers), their convergence rates are independent of the mesh size used in the discretization. The actual costs of using them depends on both the convergence rate and the cost per iteration. Our goal in this paper is to compare costs, in operation counts, of using each of the methods to solve three discrete versions of (1). For convergence to be independent of mesh size, the first three methods (*Krylov subspace methods*) require a preconditioning operator spectrally equivalent to the discrete Laplacian. In an effort to unify the comparison of these ideas with multigrid, we also implement this preconditioner using a multigrid method for the associated Poisson equation. (Thus, the Krylov subspace methods can themselves be viewed as variants of multigrid.) Our main conclusions are as follows. For problems where it is applicable, one version of multigrid, using incomplete factorization, requires the fewest iterations and operations, but it is only marginally faster, i.e., by factors of approximately 1.5 to 2, than the Krylov subspace methods and the distributed Gauss-Seidel method. The Krylov subspace methods are more widely applicable than either multigrid method. Among the Krylov subspace methods, the conjugate residual method is slightly slower than the conjugate gradient method and in some cases the Uzawa method, but it has the advantage of not requiring any parameter estimates.

An outline of the rest of the paper is as follows. In §2, we present the solution algorithms and give an overview of their convergence properties. In §3, we specify four benchmark problems and the computational costs per iteration of each of the solution methods. In §4, we present the numerical comparison.

2. Overview of methods. In this section, we present the four algorithms under consideration and outline their convergence properties. The first three methods

depend on a preconditioning operator Q_A that approximates the matrix A of (2). We assume that Q_A is symmetric positive definite (SPD) and that

$$\eta_1 \leq \frac{(v, Av)}{(v, Q_A v)} \leq \eta_2 , \tag{3}$$

where η_1 and η_2 are independent of the mesh size used in the discretization. In addition, finite element discretizations of (1) have a mass matrix M associated with the pressure discretization.[1] The preconditioner will also include a SPD approximation Q_M of M. Discussions of computational costs will be made in terms of various matrix operations together with inner products and "AXPY's," i.e., vector operations of the form $y \leftarrow \alpha x + y$.

2.1. The inexact Uzawa method. We use the following "inexact" version of the Uzawa algorithm [11], which starts with $u_0 \equiv 0$ and an arbitrary initial guess p_0:

$$\begin{aligned} &\texttt{for } i = 0 \texttt{ until convergence, do} \\ &\quad u_{i+1} = u_i + Q_A^{-1}(f - (Au_i + B^T p_i)) \\ &\quad p_{i+1} = p_i + \alpha Q_M^{-1}(Bu_{i+1} - Cp_i) \\ &\texttt{enddo} \end{aligned} \tag{4}$$

Here, α is a scalar parameter that must be determined prior to the iteration.

In the "exact" version of this algorithm, $Q_A = A$ and the first step is equivalent to solving the linear system $Au_{i+1} = f - B^T p_i$. When $Q_M = I$, the exact algorithm is then a fixed parameter first order Richardson iteration applied to the Schur complement system $(BA^{-1}B^T + C)p = BA^{-1}f$; Q_M is a preconditioner for this iteration. The inexact Uzawa algorithm (4) replaces the exact computation of $A^{-1}(f - B^T p_i)$ with an approximation.

2.2. A preconditioned conjugate gradient method. Let $\mathcal{A}$ denote the coefficient matrix of (2). Premultiplication of (2) by the matrix

$$\mathcal{T} = \begin{pmatrix} Q_A^{-1} & 0 \\ BQ_A^{-1} & -I \end{pmatrix}$$

produces the equivalent system

$$\begin{pmatrix} Q_A^{-1}A & Q_A^{-1}B^T \\ BQ_A^{-1}A - B & BQ_A^{-1}B^T + C \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} Q_A^{-1}f \\ BQ_A^{-1}f \end{pmatrix}. \tag{5}$$

Let $\mathcal{M} = \mathcal{T}\mathcal{A}$ denote the coefficient matrix of this system. The conjugate gradient method (CG) developed in [5] requires that the bilinear form

$$\left[\begin{pmatrix} v_1 \\ q_1 \end{pmatrix}, \begin{pmatrix} v_2 \\ q_2 \end{pmatrix} \right] \equiv ((A - Q_A)v_1, v_2) + (q_1, q_2) \tag{6}$$

[1] If the finite element solution is expressed using a given basis $\{\phi_i\}$ as $p = \sum_i \delta_i \phi_i$, then $\|p\|_{L_2} = (\delta, M\delta)^{1/2}$.

define an inner product. Equivalently, the preconditioning operator Q_A must satisfy (3) with $\eta_1 > 1$. It is shown in [5] that $\mathcal{M}$ is SPD with respect to the inner product (6), so that CG in this inner product is applicable. The matrix

$$\mathcal{G} = \begin{pmatrix} I & 0 \\ 0 & Q_M \end{pmatrix} \tag{7}$$

is also SPD with respect to (6), so that this can be used as a preconditioner.

Let

$$X_0 = \begin{pmatrix} u_0 \\ p_0 \end{pmatrix}, \quad R_0 = \begin{pmatrix} f - (Au_0 + B^T p_0) \\ -(Bu_0 - Cp_0) \end{pmatrix}$$

denote an arbitrary guess for the solution and the associated residual. An implementation of PCG is given below. Except for the nonstandard inner product, it is the standard implementation, as given for example in [15, p. 529]. It is more efficient than the version given in [5]. The preconditioner Q_A is implicitly incorporated into the inner product. The use of a preconditioner (7) is new.

$$\begin{aligned}
&\hat{R}_0 = \mathcal{T} R_0, \quad \tilde{R}_0 = \mathcal{G}^{-1}\hat{R}_0 \\
&P_0 = \tilde{R}_0, \quad \mathcal{M}P_0 = \mathcal{T}\mathcal{A}P_0 \\
&\alpha_0^{(n)} = [\hat{R}_0, \tilde{R}_0], \quad \alpha_0^{(d)} = [P_0, \mathcal{M}P_0], \quad \alpha_0 = \alpha_0^{(n)}/\alpha_0^{(d)} \\
&X_1 = X_0 + \alpha_0 P_0 \\
&R_1 = R_0 - \alpha_0 \mathcal{A}P_0, \quad \hat{R}_1 = \hat{R}_0 - \alpha_0 \mathcal{M}P_0, \quad \tilde{R}_1 = \mathcal{G}^{-1}\hat{R}_1 \\
&\texttt{for } i = 1 \texttt{ until convergence, do} \\
&\quad \beta_{i-1}^{(n)} = [\hat{R}_i, \tilde{R}_i], \quad \beta_{i-1}^{(d)} = \alpha_{i-1}^{(n)}, \quad \beta_{i-1} = \beta_{i-1}^{(n)}/\beta_{i-1}^{(d)} \\
&\quad P_i = \tilde{R}_i + \beta_{i-1} P_{i-1}, \quad \mathcal{M}P_i = \mathcal{T}\mathcal{A}P_i \\
&\quad \alpha_i^{(n)} = \beta_{i-1}^{(n)}, \quad \alpha_i^{(d)} = [P_i, \mathcal{M}P_i], \quad \alpha_i = \alpha_i^{(n)}/\alpha_i^{(d)} \\
&\quad X_{i+1} = X_i + \alpha_i P_i, \quad R_{i+1} = R_i - \alpha_i \mathcal{A}P_i \\
&\quad \hat{R}_{i+1} = \hat{R}_i - \alpha_i \mathcal{M}P_i, \quad \tilde{R}_{i+1} = \mathcal{G}^{-1}\hat{R}_{i+1} \\
&\texttt{enddo}
\end{aligned}$$

To help identify operation counts, we describe the computation of $\{\alpha_i\}$ and $\{\beta_i\}$ in more detail. Letting

$$R_i = \begin{pmatrix} r_i \\ s_i \end{pmatrix}, \ \hat{R}_i = \begin{pmatrix} \hat{r}_i \\ \hat{s}_i \end{pmatrix}, \ \tilde{R}_i = \begin{pmatrix} \tilde{r}_i \\ \tilde{s}_i \end{pmatrix},$$

we have $\beta_{i-1}^{(n)} = [\hat{R}_i, \tilde{R}_i] = (\hat{r}_i, A\tilde{r}_i - r_i) + (\hat{s}_i, \tilde{s}_i)$; similarly, if

$$P_i = \begin{pmatrix} c_i \\ d_i \end{pmatrix}, \ \mathcal{A}P_i = \begin{pmatrix} v_i \\ w_i \end{pmatrix}, \ \mathcal{M}P_i = \begin{pmatrix} Q_A^{-1} v_i \\ BQ_A^{-1} v_i - w_i \end{pmatrix}, \tag{8}$$

then $\alpha_i^{(d)} = [P_i, \mathcal{M}P_i] = (c_i, AQ_A^{-1}v_i - v_i) + (d_i, BQ_A^{-1}v_i - w_i)$. Q_A is referenced only in the construction of $Q_A^{-1}v$ in (8), so that only the action of the inverse of Q_A is required. Moreover, although the vectors $A\tilde{r}_i$, Ac_i (for v_i) and $AQ_A^{-1}v_i$ are used, the first two of these can be computed using an AXPY. Consequently, only one matrix-vector product by A is needed.

2.3. The preconditioned conjugate residual method. Since $\mathcal{A}$ is symmetric, variants of the conjugate residual method are applicable. Let X_0 denote the initial guess and R_0 its residual. The following algorithm implements the ORTHOMIN version of PCR with preconditioner $\mathcal{Q}$ [3]:[2]

$\tilde{R}_0 = \mathcal{Q}^{-1}R_0,\ P_0 = \tilde{R}_0,\ S_0 = \mathcal{Q}^{-1}\mathcal{A}P_0$
$\alpha_0^{(n)} = (\tilde{R}_0, \mathcal{A}P_0),\ \alpha_0^{(d)} = (\mathcal{A}P_0, S_0),\ \alpha_0 = \alpha_0^{(n)}/\alpha_0^{(d)}$
$X_1 = X_0 + \alpha_0 P_0,\ R_1 = R_0 - \alpha_0 \mathcal{A}P_0,\ \tilde{R}_1 = \tilde{R}_0 - \alpha_0 S_0$
for $i = 1$ **until** convergence, **do**
  $\beta_{i-1}^{(n)} = -(\mathcal{A}\tilde{R}_i, S_{i-1}),\quad \beta_{i-1}^{(d)} = \alpha_{i-1}^{(d)}$
  $P_i = \tilde{R}_i + \beta_{i-1}P_{i-1},\quad \mathcal{A}P_i = \mathcal{A}\tilde{R}_i + \beta_{i-1}\mathcal{A}P_{i-1},\quad S_i = \mathcal{Q}^{-1}\mathcal{A}P_i$
  $\alpha_i^{(n)} = (\tilde{R}_i, \mathcal{A}P_i),\quad \alpha_i^{(d)} = (\mathcal{A}P_i, S_i),\quad \alpha_i = \alpha_i^{(n)}/\alpha_i^{(d)}$
  $X_{i+1} = X_i + \alpha_i P_i,\quad R_{i+1} = R_i - \alpha_i \mathcal{A}P_i,\quad \tilde{R}_{i+1} = \tilde{R}_i - \alpha_i S_i$
enddo

Any symmetric positive-definite $\mathcal{Q}$ could be used as a preconditioner. As in [26], we use

$$\mathcal{Q} = \begin{pmatrix} Q_A & 0 \\ 0 & Q_M \end{pmatrix}.$$

2.4. Multigrid. As is well known, multigrid methods combine iterative methods to smooth the error with correction derived from a coarse grid computation. We use V-cycle multigrid for "transformed systems." Our description follows [34, 35]. Compare [22, 30] for other multigrid methods derived from the squared system associated with (2).

Let $-\Delta_p$ denote the Laplace operator defined on the pressure space, with Neumann boundary conditions (see [16]), and let A_p be a discrete approximation to $-\Delta_p$ defined on the pressure grid. Consider the following transformed version of (2):

$$(9)\qquad \begin{pmatrix} A & B^T \\ B & -C \end{pmatrix} \begin{pmatrix} I & B^T \\ 0 & -A_p \end{pmatrix} \begin{pmatrix} \hat{u} \\ \hat{p} \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix}, \qquad \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} I & B^T \\ 0 & -A_p \end{pmatrix} \begin{pmatrix} \hat{u} \\ \hat{p} \end{pmatrix}.$$

The coefficient matrix in (9) is

$$(10)\qquad \tilde{A} = \begin{pmatrix} A & W \\ B & G \end{pmatrix},$$

where $W = AB^T - B^T A_p$ and $G = BB^T + CA_p$. For appropriate discretizations of (1) (see §3), W is of low rank, with nonzero entries only in rows corresponding to mesh points next to $\partial\Omega$. When $C = 0$, G can also be viewed as discretization of $-\Delta_p$. The splitting

$$(11)\qquad \tilde{A} = \mathcal{S} - \mathcal{R}$$

[2] It is possible for this version of PCR to break down, with $\alpha_i = 0$. The ORTHODIR version, which uses a three-term recurrence to generate P_i, is guaranteed not to break down; it requires two additional AXPY's. Our implementation switches from the ORTHOMIN to ORTHODIR direction update if $|\alpha_i| < 10^{-4}$, as described in [9]. In the experiments discussed in §4, this switch never took place.

then induces a stationary iteration applicable to (2),

$$\begin{pmatrix} u_{k+1} \\ p_{k+1} \end{pmatrix} = \begin{pmatrix} u_k \\ p_k \end{pmatrix} + \begin{pmatrix} I & B^T \\ 0 & -A_p \end{pmatrix} \mathcal{S}^{-1} \begin{pmatrix} f - (Au_k + B^T p_k) \\ -(Bu_k - Cp_k) \end{pmatrix}. \tag{12}$$

This is used as the smoother for the multigrid solver for (2). Specific choices for $\mathcal{S}$ are given in §3.2.

Let R_u denote a restriction operator mapping velocity vectors in the fine grid (of width h) to the coarse grid (of width $2h$), let R_p similarly denote the restriction operator for the discrete pressure space, and let P_u and P_p denote prolongation operators from the coarse spaces to the fine spaces. (For simplicity, we are omitting explicit mention of h in this notation.) One step of V-cycle multigrid for solving (2), starting with initial guess u^0, p^0, is as follows.

$(u^1, p^1) = \mathbf{MG}(u^0, p^0, f, g, k_1, k_2, h)$
if $h < h_0$, **then** % Recursive call
  Starting with u^0, p^0, perform k_1 smoothing steps (12), producing $u^{1/3}, p^{1/3}$
  $r^{1/3} = f - (Au^{1/3} + B^T p^{1/3})$, $\quad s^{1/3} = -(Bu^{1/3} - Cp^{1/3})$
  $r_c^{1/3} = R_u r^{1/3}$, $\quad s_c^{1/3} = R_p s^{1/3}$
  $(u_c^{2/3}, p_c^{2/3}) = \mathbf{MG}(0, 0, r_c^{1/3}, s_c^{1/3}, k_1, k_2, 2h)$
  $u^{2/3} = u^{1/3} + P_u u_c^{2/3}$, $\quad p^{2/3} = p^{1/3} + P_p p_c^{2/3}$
  Starting with $u^{2/3}, p^{2/3}$, perform k_2 smoothing steps (12), producing u^1, p^1
else % Coarse grid solve when $h = h_0$
  Solve $\begin{pmatrix} A & B^T \\ B & -C \end{pmatrix} \begin{pmatrix} u^1 \\ p^1 \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix}$ directly
endif

We also use V-cycle multigrid derived from the discrete Laplacian as a preconditioner to approximate the action of A^{-1} for the Krylov subspace methods; this is defined analogously and we omit the details. For all multigrid methods, we use bilinear interpolation to define P_u and P_p, and $R_u = P_u^T$, $R_p = P_p^T$. The discrete operators at each level are derived from the discretization on the associated grid.

2.5. Convergence properties. We briefly outline some convergence properties of these methods; see the primary references for derivations of bounds. Each of the methods generates a sequence of iterates $u_i \approx u$, $p_i \approx p$ such that, if e_i is a representation of the error, then $\lim_{i\to\infty}(\|e_i\|/\|e_0\|)^{1/i} = \rho$ for some norm $\|\cdot\|$. We refer to ρ as the *convergence factor.*

We are assuming that the discretization and choice of Q_M are such that

$$\lambda_1 \le \frac{(q, (BA^{-1}B^T + C)q)}{(q, Q_M q)} \le \lambda_2, \tag{13}$$

where λ_1 and λ_2, and therefore, $\kappa \equiv \lambda_2/\lambda_1$, are bounded independently of the mesh size of the discretization. This is the case, for example, when Q_M is a suitable approximation of the mass matrix in finite element discretization [29, 32]. Note that κ is the spectral condition number of $Q_M^{-1}(BA^{-1}B^T + C)$.

The exact Uzawa algorithm has convergence factor $\rho\left(I - \alpha Q_M^{-1}(BA^{-1}B^T + C)\right)$ [12]. This is smallest for the choice $\alpha = 2/(\lambda_1 + \lambda_2)$, in which case it has the value $(\kappa - 1)/(\kappa + 1)$. Thus, the convergence factor for the Uzawa algorithm is independent of the mesh. It is shown in [11] that the performance of the inexact Uzawa algorithm is close to that of the exact one if the iterate u_{i+1} satisfies

$$\|f - B^T p_i - A u_{i+1}\|_2 < \tau \|B u_i - C p_i\|_{Q_A^{-1}}, \tag{14}$$

where τ is independent of the mesh size.

The PCG method is analyzed in [5, Theorem 1], where it is shown that the condition number of the coefficient matrix $\mathcal{M}$ of (5) is bounded by a constant proportional to κ. Thus, standard results for CG [15] imply that the bound on the convergence factor for this method is proportional to $(\sqrt{\kappa} - 1)/(\sqrt{\kappa} + 1)$. The constant of proportionality depends on how close η_1 is to η_2 in (3), i.e., how well Q_A approximates A.

The PCR method is analyzed in [24, 26]. The analysis shows that the eigenvalues of the preconditioned matrix $\mathcal{Q}^{-1}\mathcal{A}$ are contained in two intervals $[-a, -b] \cup [c, d]$, where a, b, c, are d are positive constants that are independent of the mesh size. The sizes of the intervals depend on κ and the accuracy with which Q_A approximates A. It follows from the convergence analysis of CR [9, 27] that the convergence factor for the preconditioned algorithm is independent of the mesh size. For example, it is shown [9] that if $d - c = a - b > 0$, then the convergence factor is bounded by $2\left(\frac{1-\sqrt{\beta}}{1+\sqrt{\beta}}\right)^{1/2}$, where $\beta = (bc)/(ad)$.

It is shown in [36] that for finite difference discretization of (1) (see §3.1), two-grid variants of multigrid are convergent with convergence rate independent of the mesh size. The analysis applies to the ILU smoothing of §3.2, although it requires that the prolongation be based on biquadratic interpolation. In practice, bilinear interpolation has been observed to be sufficient [35]. Fourier analysis in [6] also suggests that MG/DGS has convergence rate independent of mesh size.

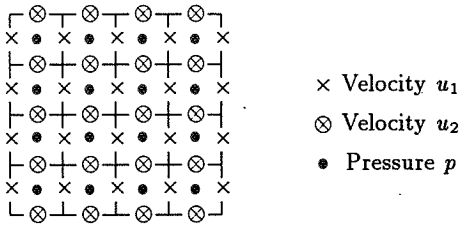
REMARK 2.1. Several other proposed methods share properties with the version of PCG under consideration. In particular, Verfürth [29] has shown that PCG applied directly to the Schur complement system has convergence factor proportional to ρ_{CG}; however, this method requires accurate computation of the action of A^{-1} at each CG step [23]. Bank, Welfert, and Yserentant [4] present a method making use of $Q_A \approx A$ with convergence rate dependent on the accuracy of this approximation, but using an additional inner iteration on the pressure space.

3. Solution costs. In this section, we outline the computational costs required to solve three benchmark problems on $\Omega = (0,1) \times (0,1)$ for each of the solution methods of §2.

3.1. Benchmark problems. We use four discretizations to produce test problems: "marker and cell" finite differences and three mixed finite element strategies.

1. *Finite differences* [19]. This consists of the usual five-point operator for each of the discrete Laplacian operators of (1), together with centered differences for the first derivatives ∇p and $\operatorname{div} u$. For the discretization to be stable, it is necessary to use

FIG. 1. *Staggered grids for finite difference discretization.*



staggered grids in $\bar{\Omega}$. Figure 1 shows such grids on a mesh of width $h = 1/4$. In order to define the velocity discretizations at grid points next to $\partial\Omega$, certain values outside $\bar{\Omega}$ must be extrapolated; for example, this is needed to approximate $\partial^2 u_1/\partial y^2$ for points "×" next to the bottom of $\partial\Omega$.

2. *Linear/constant finite elements.* This choice consists of continuous piecewise linear velocities on a mesh of width h, and piecewise constant pressures on a mesh of width $2h$. The discrete pressures are not required to be continuous. The coarser pressure grid ensures that the inf-sup condition holds [17]. We refer to this as the $P_1(h)P_0(2h)$ discretization.

3. *Piecewise linear finite elements.* Here, continuous piecewise linear velocities on a mesh of width h are paired with continuous piecewise linear pressures on a mesh of width $2h$. The inf-sup condition is also satisfied. We call this the $P_1(h)P_1(2h)$ discretization.

4. *Stabilized piecewise linear finite elements.* A stable discretization using piecewise linear velocities and pressures on a single of mesh can be obtained using a stabilization matrix $C = \beta h^2 A_n$, where A_n is the discrete Laplace operator defined on the pressure space, subject to Neumann boundary conditions [8]. This technique is equivalent to the mini-element discretization [1] after elimination of the internal degrees of freedom. We use $\beta = .025$, as recommended in [25]. We refer to this discretization as $P_1(h)P_1(h)$. The usual hat functions are used as the bases for linear velocities and pressures.

The coefficient matrix $\mathcal{A}$ of (2) for all these problems, as well as B^T, C, and $BA^{-1}B^T + C$, are rank deficient by one; the latter three matrices share a constant null vector. As a result, the discrete pressure solutions are uniquely defined only up to a constant. In exact arithmetic, the solution methods under consideration correct the initial guess with quantities orthogonal to the null space of $\mathcal{A}$, so that the component of the null space in the computed solution is the same as in the initial guess. For the analysis, the lower bound of (13) refers to the smallest nonzero eigenvalue.

Note that our goal in considering these problems is to compare the performance of the different solution strategies on a variety of problems. We highlight some properties of each of the problems as follows:

1. finite differences, stable, #(pressure unknowns) ≈ #(velocity grid points);
2. finite elements, stable, discontinuous pressures, #(pressure unknowns) ≈ $\frac{1}{2}$ #(velocity grid points);
3. finite elements, stable, continuous pressures, #(pressure unknowns) ≈ $\frac{1}{4}$ #(velocity grid points);
4. finite elements, requires stabilization, continuous pressures, #(pressure un-

knowns) $\approx$ #(velocity grid points).

We are not comparing the accuracy achieved by the discretizations, and remark only that the three finite element discretizations display the same asymptotic convergence rates. See [17, pp. 29, 50] for comments on accuracy of finite element discretization, and [21] for analysis of the finite difference scheme.

3.2. Preconditioners and smoothers. The Uzawa, PCR, and PCG methods require choices of Q_A and Q_M. For all cases, Q_A consists of one step of V-cycle multigrid derived from the discrete Laplacian. The smoothing is based on damped point-Jacobi iteration with damping parameter $\omega = 2/3$ [20], which ensures that Q_A is symmetric. For the three finite element discretizations, Q_M is chosen to be the diagonal of the mass matrix M; see [32]. (In the case of the $P_1(h)P_0(2h)$ discretization, $Q_M = M$.) Although there is no mass matrix for finite differences, a natural analogue in two dimensions is $M = h^2 I$, and this is used for Q_M with finite differences.

We consider two multigrid smoothing strategies. The first is a variant of the distributed Gauss-Seidel (DGS) iteration introduced by Brandt and Dinar [6]. The splitting operator of (11) is given by

$$\mathcal{S} = \begin{pmatrix} S_A & 0 \\ B & S_G \end{pmatrix},$$

so that the smoother (12) has the form

$$\begin{aligned} \tilde{u}_{k+1} &= S_A^{-1}(f - (Au_k + B^T p_k)) \\ \tilde{p}_{k+1} &= S_G^{-1}(-(B(u_k + \tilde{u}_{k+1}) + Cp_k) \\ u_{k+1} &= u_k + \tilde{u}_{k+1} + B^T \tilde{p}_{k+1} \\ p_{k+1} &= p_k - A_p \tilde{p}_{k+1}\,. \end{aligned}$$

For S_A, we use the point Gauss-Seidel matrix derived from red-black ordering of the velocity grid. (That is, if $A = D - L - U$ with the red-black ordering, then $S_A = D - L$.) For finite differences, $S_G = (1/\omega)T$ where T is the tridiagonal part of G and $\omega = 2/3$; that is, S_G corresponds to a damped one-line Jacobi splitting. For $P_1(h)P_1(h)$ finite elements, S_G is the block Jacobi matrix derived from a two-line ordering of the underlying grid. These are slightly more sophisticated versions of the choice $S_G = diag(G)$ used in [6]. We refer to this multigrid method as MG/DGS.

The other multigrid smoother is the incomplete LU factorization (ILU) presented by Wittum [35]. We use an ILU factorization of the matrix $\tilde{A}$ of (10), with no fill-in in the factors. The ordering for $\tilde{A}$ is problem dependent. For finite differences, it is derived from an uncoupled red-black ordering of the underlying grid. That is, the grid values for u_1 were listed first, in red-black ordering, followed by those for u_2, and then those for p. (See also Remark 3.3 below.) For $P_1(h)P_1(h)$ finite elements, $\tilde{A}$ is ordered according to an uncoupled *lexicographic* ordering of the grid vectors. We denote this method by MG/ILU.

In choosing preconditioners and smoothers, we have attempted to use methods that are suitable for vector and parallel computers. Thus, we are using point Jacobi smoothing for multigrid preconditioning, red-black Gauss-Seidel and line Jacobi for the DGS iteration, and a red-black ordering for MG/ILU applied to finite differences.

With the $P_1(h)P_1(h)$ discretization, the operator G in the DGS method is a 19-point operator that has block Property A for a two-line ordering of the pressure grid, so that the two-line Jacobi splitting can be implemented efficiently in parallel. The ILU smoother used with this problem is not efficient on parallel computers. Our multigrid strategies do not address the issue of idleness of parallel processors for coarse grid computations; see [10, 13] for discussions of this point for the discrete Poisson equation.

Parameters are required for the Uzawa, PCG and multigrid methods, and for the multigrid preconditioner. These are as follows:

UZAWA: The optimal value of α for the exact Uzawa method, determined empirically, is used for the inexact version. This requires computation of the extreme eigenvalues of $Q_M^{-1}(BA^{-1}B^T + C)$.

PCG: As noted in §2.3, the preconditioner must be scaled so that $\eta_1 > 1$ in (3). From the results of [5], it is desirable to have η_1 close to 1. In all tests, the scaling is chosen so that $1 < \eta_1 < 1.02$. This requires computation of the smallest eigenvalue of $Q_A^{-1}A$.[3]

MULTIGRID: For the coarse mesh size h_0 in multigrid computations, we chose the one of $h_0 = 1/2$ and $h_0 = 1/4$ that produced lower iteration counts. This turned out to be $h_0 = 1/2$ for preconditioners and $h_0 = 1/4$ for solvers. The coarse grid solution is obtained using Cholesky factorization for the preconditioners and singular value decomposition for the solvers.

REMARK 3.1. For the Uzawa method, the choice of Q_A does not guarantee that the condition (14) is satisfied. The results of [11, 33] as well as those of §4 suggest that with multigrid for Q_A, (14) may be too stringent.

REMARK 3.2. The effectiveness of the multigrid solvers depends on the fact that the commutator W in (10) is zero away from the boundary of Ω. This is true for the finite difference and stabilized $P_1(h)P_1(h)$ discretizations, where pressures and velocities are defined on the same grid, but not for the (stable) $P_1(h)P_1(2h)$ discretization. Our experiments confirm that multigrid is ineffective for this discretization, and we do not include it as an option. See [18, p. 248] for a discussion of this issue. For the $P_1(h)P_0(2h)$ discretization, it is difficult to define the discrete pressure Poisson operator A_p, and we know of no multigrid implementation for this problem.

REMARK 3.3. For MG/ILU applied to the finite difference discretization, we also tested several alternative ordering strategies, including an uncoupled lexicographic ordering (i.e., like that used for $P_1(h)P_1(h)$), as well as several "coupled" lexicographic orderings. For the latter strategies, velocity and pressure unknowns are not separated from one another, see [28]. The performances of MG/ILU for all these orderings were very close. For example, for $h = 1/32$ as in Table 4 below, the smallest average iteration count with one smoothing step was $10\frac{1}{3}$ and the largest was $11\frac{2}{3}$.

3.3. Iteration costs. We identify the costs per iteration of each of the methods by first specifying the "high level" operations of which they are composed, and then determining the costs of each of these operations. High level operations are defined to be matrix-vector products, inner products (denoted "(,)" in the tables of this section), and AXPY's. Note that each of the techniques under consideration is formulated with

[3] In the experiments described in §4, these were computed using a power method applied to $Q_A^{-1}A - I$; five to ten steps were needed to obtain an estimate accurate to three significant digits.

TABLE 1
High level operations for all solution algorithms.

	Matrix-Vector Product			AXPY	(,)
Uzawa	$1\ A$	$1\ B^T$	$1\ Q_A^{-1}$	$1\ (n_p)$	$1\ (n_u + n_p)$
	$1\ B$	$1\ C$	$1\ Q_M^{-1}$		
PCG	$1\ A$	$1\ B^T$	$1\ Q_A^{-1}$	$4\ (n_u + n_p)$	$3\ (n_u + n_p)$
	$2\ B$	$1\ C$	$1\ Q_M^{-1}$	$2\ (n_u)$	
PCR	$1\ A$	$1\ B^T$	$1\ Q_A^{-1}$	$5\ (n_u + n_p)$	$4\ (n_u + n_p)$
	$1\ B$	$1\ C$	$1\ Q_M^{-1}$		
Multigrid	$(1 + k_1 + k_2)\ A$		$1\ R_u$		
Preconditioner	$(k_1 + k_2)\ S_A^{-1}$		$1\ P_u$		
Multigrid Solver	$1\ A$	$1\ B^T$	$1\ R_u$		$1\ (n_u + n_p)$
(Excluding	$1\ B$	$1\ C$	$1\ R_p$		
smoother)	$1\ P_u$	$1\ P_p$			
	$1\ A$	$2\ B^T$	$1\ A_p$		
DGS Smoother	$1\ B$	$1\ C$	$1\ S_A^{-1}$		
			$1\ S_G^{-1}$		
ILU Smoother	$1\ A$	$2\ B^T$	$1\ A_p$		
	$1\ B$	$1\ C$	$1\ \mathcal{S}^{-1}$		

essentially the same set of these operations; consequently, we expect operation counts to give a good idea of their comparative performance.

The high level operations are shown in Table 1. Matrix-vector products include operations with matrices that define the problem or method, such as A or R_u, as well as preconditioning and smoothing operators such as Q_A^{-1} and S_A^{-1}. The latter computations are themselves built from other matrix operations, and some of these are also identified in the table. All multigrid entries correspond to operations performed on one grid level. For multigrid solvers, the smoothing operations are presented separately; these operations would be performed k_1 times during presmoothing and k_2 times during postsmoothing. The lengths of the vector operations are listed in parentheses. We are assuming that one inner product will be used in the convergence test, and the counts in the table include this.

The costs of matrix-vector products are estimated to be the number of nonzeros in the matrices used. This is roughly one half the number of "FLOPS" required, and it is also proportional to the number of memory references. These costs, for discretizations in which the velocity unknowns come from an $n \times n$ grid, are shown in Table 2. The costs of vector operations are taken to be the length of the vectors.

Combining the data of Tables 1 and 2 gives an estimate for the cost per iteration for each of the solution methods under consideration. These numbers are all proportional to n^2, and we present in Table 3 the cost factors obtained by omitting this factor, rounded to the nearest integer. For the multigrid methods (preconditioners and solvers), the cost of one full multigrid step is estimated as $4/3$ times the cost of the computations on the finest grid; this is approximately the cost of full recursive multigrid in two dimensions.

TABLE 2
Costs for matrix-vector products.

	Fin. Diff.	$P_1(h)P_0(2h)$	$P_1(h)P_1(2h)$	$P_1(h)P_1(h)$
A	$10n^2$	$10n^2$	$10n^2$	$10n^2$
B, B^T	$4n^2$	$4n^2$	$8n^2$	$12n^2$
C	0	0	0	$5n^2$
Q_M^{-1}	$1n^2$	$0.25n^2$	$0.25n^2$	$1n^2$
S_A^{-1} (Jacobi)	$2n^2$	$2n^2$	$2n^2$	$2n^2$
S_A^{-1} (Gauss-Seidel)	$6n^2$	$6n^2$	$6n^2$	$6n^2$
S_G^{-1}	$3n^2$	–	–	$9n^2$
A_p	$5n^2$	–	–	$5n^2$
R_u, P_u	$6n^2$	$4.5n^2$	$4.5n^2$	$4.5n^2$
R_p, P_p	$3n^2$	–	–	$2.25n^2$
$\mathcal{S}^{-1}$	$19n^2$	–	–	$41n^2$

TABLE 3
Cost factors.

		Uzawa	PCR	PCG	MG/DGS	MG/IC
Finite	$k_1 = k_2 = 1$	84	107	109	148	175
Differences	$k_1 = k_2 = 2$	116	139	141	244	297
$P_1(h)P_0(2h)$	$k_1 = k_2 = 1$	79	98	101	–	–
	$k_2 = k_2 = 2$	111	130	133	–	–
$P_1(h)P_1(2h)$	$k_1 = k_2 = 1$	86	104	111	–	–
	$k_2 = k_2 = 2$	118	136	143	–	–
$P_1(h)P_1(h)$	$k_1 = k_2 = 1$	101	124	134	247	333
	$k_2 = k_2 = 2$	133	156	166	421	591

TABLE 4
Iterations.

		Uzawa	PCR	PCG	MG/DGS	MG/ILU
Finite	$k_1 = k_2 = 1$	36	41	30	24	12
Differences	$k_1 = k_2 = 2$	28	33	23	15	9
$P_1(h)P_0(2h)$	$k_1 = k_2 = 1$	34	41	29	–	–
	$k_2 = k_2 = 2$	26	34	23	–	–
$P_1(h)P_1(2h)$	$k_1 = k_2 = 1$	89	57	38	–	–
	$k_2 = k_2 = 2$	89	50	31	–	–
$P_1(h)P_1(h)$	$k_1 = k_2 = 1$	39	47	32	20	8
	$k_1 = k_2 = 2$	38	40	25	10	7

4. Experimental results. We now present the results of numerical experiments for solving (2). All experiments were performed in MATLAB on a SPARC-10 workstation. For each solution algorithm, we solved three problems derived from three choices of f consisting of uniformly distributed random numbers in $[-1, 1]$. The initial guess in all cases was $u_0 = 0$, $p_0 = 0$. The stopping criterion was

$$\|R_i\|_2/\|R_0\|_2 < 10^{-6},$$

where

$$R_i = \begin{pmatrix} f \\ 0 \end{pmatrix} - \begin{pmatrix} A & B^T \\ B & -C \end{pmatrix} \begin{pmatrix} u_i \\ p_i \end{pmatrix}.$$

We found that performance was essentially in the asymptotic range for $h = 1/32$, and all results are for this mesh size.

We present three types of data: iteration counts, estimates for convergence factors, and plots of residual norms as functions of operation counts. The iteration counts are averages over three runs of the number of steps needed to satisfy the stopping criterion; these are shown in Table 4. The estimates for asymptotic convergence factors are the averages of $(\|\bar{R}_{5+i}\|_2/\|\bar{R}_5\|_2)^{1/i}$ over all steps after step five; here $\bar{R}_k$ represents the average of the kth residual norm over the three runs. These are shown in Table 5. We chose step five rather than step zero because performance was often better in the first few steps than later, when the asymptotic behavior is seen. Finally, Figures 2 – 5 plot the averages of the residual norms against operation counts.

We make the following observations on these results.

1. Where it is applicable, multigrid requires the smallest number of iterations and has the smallest convergence factors. MG/ILU is superior to MG/DGS in these measures. These observations agree with those of [35]. In addition, where it is applicable, MG/ILU requires the smallest number of operations. However, multigrid is only effective for discretizations where velocities and pressures are defined on the same grid.
2. The Krylov subspace methods and MG/DGS are roughly equal in cost. The Krylov subspace methods are more widely applicable than multigrid.
3. The performances of all these methods are very close. In terms of operation counts, the ratio of costs of the most expensive and least expensive method is no worse than 2.3.

TABLE 5
Estimates of convergence factors.

		Uzawa	PCR	PCG	MG/DGS	MG/ILU
Finite	$k_1 = k_2 = 1$	.67	.70	.66	.62	.39
Differences	$k_1 = k_2 = 2$	.60	.64	.57	.50	.31
$P_1(h)P_0(2h)$	$k_1 = k_2 = 1$	.69	.69	.70	–	–
	$k_2 = k_2 = 2$	.58	.66	.55	–	–
$P_1(h)P_1(2h)$	$k_1 = k_2 = 1$	.82	.79	.75	–	–
	$k_2 = k_2 = 2$	.84	.78	.70	–	–
$P_1(h)P_1(h)$	$k_1 = k_2 = 1$	.70	.75	.68	.56	.24
	$k_1 = k_2 = 2$	.70	.74	.62	.33	.21

FIG. 2. *Operation counts for finite difference discretization.*

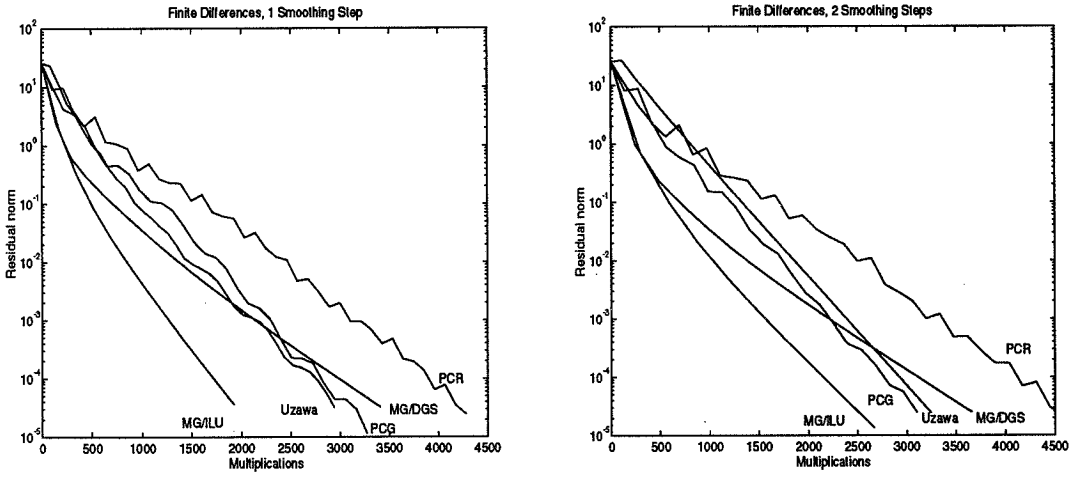


FIG. 3. *Operation counts for $P_1(h)P_0(2h)$ finite element discretization.*

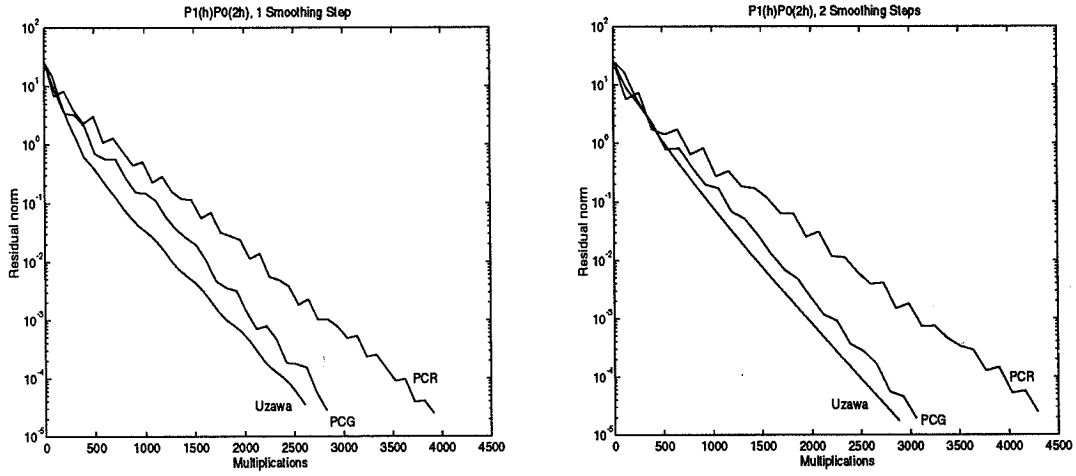


FIG. 4. *Operation counts for $P_1(h)P_1(2h)$ finite element discretization.*

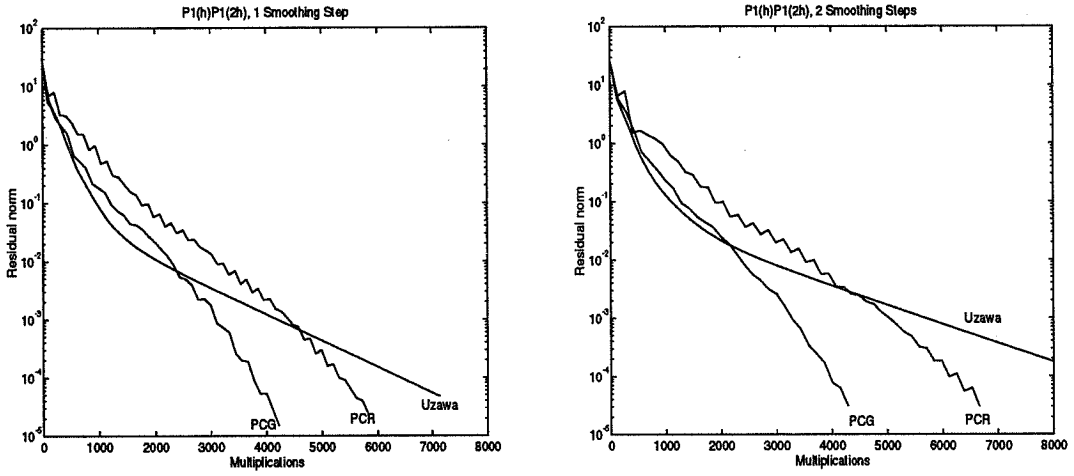
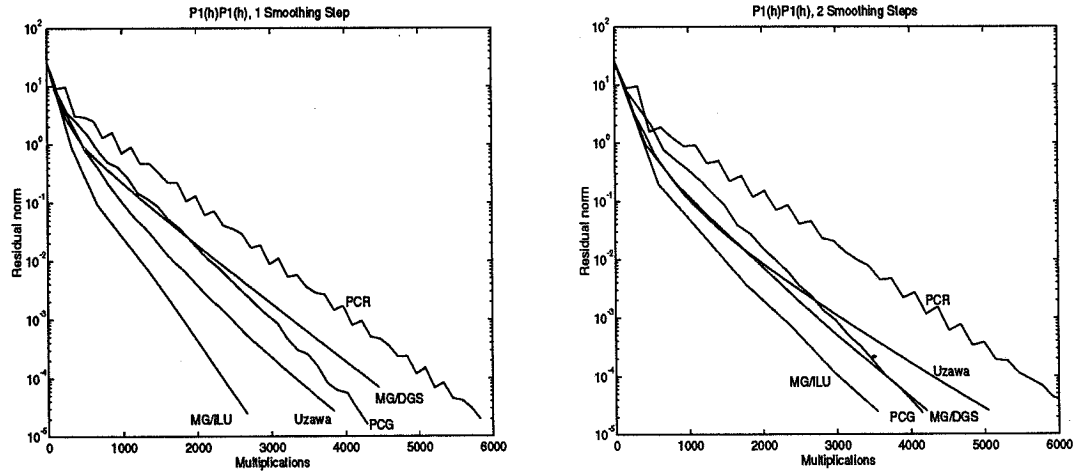


FIG. 5. *Operation counts for $P_1(h)P_1(h)$ finite element discretization.*



4. No Krylov subspace method is clearly superior to the others. PCG exhibits a somewhat faster convergence rate than PCR, and the Uzawa algorithm is surprisingly competitive with the other two methods. This appears to derive from the dependence of PCG and PCR on both the spectral condition number κ from (13) and the accuracy of the preconditioner Q_A as an approximation to A; for both these methods, the iteration counts go down in all cases when the number of smoothing steps in Q_A increases. The Uzawa method appears to be less sensitive to the accuracy of Q_A. The values of κ for the three problems are:

Finite differences	4.14	$P_1(h)P_1(2h)$	22.71
$P_1(h)P_0(2h)$	4.87	$P_1(h)P_1(h)$	9.91

The Uzawa method is least effective for the $P_1(h)P_1(2h)$ discretization, which has the largest condition number.

5. The Uzawa and PCG methods depend on choices of iteration parameters. These can be estimated relatively inexpensively (e.g., using a coarse grid for the Uzawa method, and a few steps of the power method for PCG), but this increases the cost of these methods and makes implementing them considerably more difficult. In contrast, PCR is independent of parameters except for those needed for the multigrid preconditioning, and it is therefore easier to implement. Thus, there is a tradeoff between these methodologies: PCR converges slightly more slowly than PCG and, often, than the Uzawa method, but it has a simpler implementation.

6. For each of the solution strategies except PCG, it is less expensive to use one smoothing step than two.

Acknowledgements. The author wishes to thank David Silvester for a careful reading of a preliminary version of this paper, and Andy Wathen for some helpful remarks.

REFERENCES

[1] D. Arnold, F. Brezzi, and M. Fortin, *A stable finite element for the Stokes equations*, Calcolo, 21 (1984), pp. 337–344.

[2] K. Arrow, L. Hurwicz, and H. Uzawa, *Studies in Nonlinear Programming*, Stanford University Press, Stanford, CA, 1958.

[3] S. F. Ashby, T. A. Manteuffel, and P. E. Saylor, *A taxonomy for conjugate gradient methods*, SIAM J. Numer. Anal., 27 (1990), pp. 1542–1568.

[4] R. E. Bank, B. D. Welfert, and H. Yserentant, *A class of iterative methods for solving saddle point problems*, Numer. Math., 56 (1990), pp. 645–666.

[5] J. H. Bramble and J. E. Pasciak, *A preconditioning technique for indefinite systems resulting from mixed approximations of elliptic problems*, Math. Comp., 50 (1988), pp. 1–17.

[6] A. Brandt and N. Dinar, *Multigrid solutions to elliptic flow problems*, in Numerical Methods for Partial Differential Equations, S. V. Parter, ed., Academic Press, New York, 1979, pp. 53–147.

[7] F. Brezzi and M. Fortin, *Mixed and Hybrid Finite Element Methods*, Springer-Verlag, New York, 1991.

[8] F. Brezzi and J. Pitkäranta, *On the stabilisation of finite element approximations of the Stokes problem*, in Efficient Solutions of Elliptic Systems, W. Hackbusch, ed., Braunschweig, Vieweg, 1984, pp. 11–19. Notes on Numerical Fluid Mechanics, Vol 10.

[9] R. Chandra, S. C. Eisenstat, and M. H. Schultz, *The modified conjugate residual method for partial differential equations*, in Advances in Computer Methods for Partial Differential Equations II, R. Vichnevetski, ed., IMACS, New Brunswick, 1977, pp. 13–19.

[10] N. DECKER, *Note on the parallel efficiency of the Frederickson-McBryan multigrid algorithm*, SIAM J. Sci. Stat. Comput., 12 (1991), pp. 208–220.
[11] H. C. ELMAN AND G. H. GOLUB, *Inexact and preconditioned Uzawa algorithms for saddle point problems*, Tech. Report UMIACS-TR-93-41, Institute for Advanced Computer Studies, University of Maryland, 1993. To appear in *SIAM J. Numer. Anal.*
[12] M. FORTIN AND R. GLOWINSKI, *Augmented Lagrangian Methods: Applications to the Numerical Solution of Boundary-Value Problems*, North-Holland, New York, 1983.
[13] P. O. FREDERICKSON AND O. A. MCBRYAN, *Normalized convergence rates for the PSMG method*, SIAM J. Sci. Stat. Comput., 12 (1991), pp. 221–229.
[14] R. GLOWINSKI, *Numerical Methods for Nonlinear Variational Problems*, Springer-Verlag, New York, 1984.
[15] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, second ed., The Johns Hopkins University Press, Baltimore, 1989.
[16] P. M. GRESHO AND R. L. SANI, *On pressure boundary conditions for the incompressible Navier-Stokes equations*, Int. J. Numer. Meth. Fluids, 7 (1987), pp. 1111–1145.
[17] M. GUNZBURGER, *Finite Element Methods for Viscous Incompressible Flows*, Academic Press, San Diego, 1989.
[18] W. HACKBUSCH, *Multi-Grid Methods and Applications*, Springer-Verlag, Berlin, 1985.
[19] F. H. HARLOW AND J. E. WELCH, *Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface*, The Physics of Fluids, 8 (1965), pp. 2182–2189.
[20] S. F. MCCORMICK, ed., *Multigrid Methods*, SIAM, Philadelphia, 1987.
[21] R. A. NICOLAIDES, *Analysis and convergence of the MAC scheme I*, SIAM J. Numer. Anal., 29 (1992), pp. 1579–1591.
[22] J. PITKÄRANTA AND T. SAARINEN, *A multigrid version of a simple finite element method for the Stokes problem*, Math. Comp., 45 (1985), pp. 1–14.
[23] A. RAMAGE AND A. J. WATHEN, *Iterative Solution Techniques for the Navier-Stokes Equations*, Tech. Report 93-01, School of Mathematics , University of Bristol, 1993. To appear in *Int. J. Numer. Meth. Fluids.*
[24] T. RUSTEN AND R. WINTHER, *A preconditioned iterative method for saddle point problems*, SIAM J. Matr. Anal. Appl., 13 (1992), pp. 887–904.
[25] D. SILVESTER, *Optimal low order finite element methods for incompressible flow*, Comp. Meths. Appl. Mech. Engrg., 111 (1994), pp. 357–368.
[26] D. SILVESTER AND A. WATHEN, *Fast iterative solution of stabilized Stokes systems part II: using block preconditioners*, SIAM J. Numer. Anal., 31 (1994), pp. 1352–1367.
[27] D. B. SZYLD AND O. B. WIDLUND, *Variational analysis of some conjugate gradient methods*, East-West J. of Numer. Math., 1 (1993), pp. 51–74.
[28] S. P. VANKA, *Block-implicit multigrid solution of Navier-Stokes in primitive variables*, J. Comput. Phys., 65 (1986), pp. 138–158.
[29] R. VERFÜRTH, *A combined conjugate gradient-multigrid algorithm for the numerical solution of the Stokes problem*, IMA J. Numer. Anal., 4 (1984), pp. 441–455.
[30] ———, *A multilevel algorithm for mixed problems*, SIAM J. Numer. Anal., 21 (1984), pp. 264–271.
[31] A. WATHEN AND D. SILVESTER, *Fast iterative solution of stabilized Stokes systems. Part I: Using simple diagonal preconditioners*, SIAM J. Numer. Anal., 30 (1993), pp. 630–649.
[32] A. J. WATHEN, *Realistic eigenvalue bounds for the Galerkin mass matrix*, IMA J. Numer. Anal., 7 (1987), pp. 449–457.
[33] B. D. WELFERT, *Convergence of Inexact Uzawa Algorithms for Saddle Point Problems*, tech. report, Mathematics Department, University of Arizona, 1993.
[34] P. WESSELING, *An Introduction to Multigrid Methods*, John Wiley & Sons, New York, 1992.
[35] G. WITTUM, *Multi-grid methods for the Stokes and Navier-Stokes equations*, Numer. Math., 54 (1989), pp. 543–564.
[36] ———, *On the convergence of multi-grid methods with transforming smoothers*, Numer. Math., 57 (1990), pp. 15–38.

Page intentionally left blank

A New Coarsening Operator for the Optimal Preconditioning of the Dual and Primal Domain Decomposition Methods: Application to Problems with Severe Coefficient Jumps

Charbel Farhat
Department of Aerospace Engineering Sciences
and Center for Space Structures and Controls
University of Colorado at Boulder
Boulder, CO 80309-0429, U. S. A.

Daniel Rixen
Laboratoire de Techniques Aéronautiques et Spatiales
Université de Liège
Rue Ernest Solvay, 21, B-4000 Liège, Belgium

Abstract

We present an optimal preconditioning algorithm that is equally applicable to the dual (FETI) and primal (Balancing) Schur complement domain decomposition methods, and which successfully addresses the problems of subdomain heterogeneities including the effects of large jumps of coefficients. The proposed preconditioner is derived from energy principles and embeds a new coarsening operator that propagates the error globally and accelerates convergence. The resulting iterative solver is illustrated with the solution of highly heterogeneous elasticity problems.

1. Introduction

With the advent of parallel processing, domain decomposition (DD) based iterative algorithms have become increasingly popular for the solution of finite element systems of equations. Indeed, domain decomposition provides a higher level of concurrency than global algebraic approaches, and is simpler to implement on most parallel computational platforms [ref. 1]. In general, the subdomain equations are solved using a direct skyline or sparse factorization based algorithm, and the interface problem is solved iteratively—usually, by a preconditioned conjugate gradient (PCG) algorithm (for symmetric problems). The success of such an iterative algorithm hinges on two important properties: *numerical* scalability, and *parallel* scalability. A subdomain based iterative method is said to be numerically scalable if the condition number of its corresponding interface problem does not grow or grows "weakly" with the mesh size h and the subdomain size H. For example, if the interface problem has a condition number κ that

grows asymptotically as

$$\kappa = O\left(1 + log^2\left(\frac{H}{h}\right)\right) \tag{1}$$

then, the underlying subdomain based iterative method is numerically scalable. The practical implications of a condition number such as that described in Eq. (1) are twofold:

- Suppose that a given mesh is fixed: one processor is assigned to every subdomain, and the number of subdomains is increased in order to increase parallelism. In that case, h is fixed and H is decreased. From Eq. (1), it follows that the condition number of the interface problem decreases. This implies that the number of iterations for convergence can be expected to decrease with an increasing number of subdomains.

- On most distributed memory parallel processors, the total amount of available memory increases with the number of processors. When solving a certain class of problems on such parallel hardware, it is customary to define in each processor a constant subproblem size, and to increase the total problem size with the number of processors. In such a case, h and H are decreased, but the ratio H/h is kept constant. In theory, it follows from Eq. (1) that a numerically scalable DD algorithm can solve larger problems with the same number of iterations that are required for smaller problems simply by increasing the number of subdomains.

In practice, numerical scalability is most interesting when parallel scalability can also be achieved. The latter property characterizes the ability of an implemented algorithm to deliver a larger speedup for a larger number of processors. Therefore, a subdomain based iterative method that boasts both numerical and parallel scalability is clearly an "ultimate" solution algorithm. Unfortunately, numerical scalability can be achieved only if, at each CG iteration, the DD algorithm can propagate the error globally to accelerate convergence. Since a global propagation usually induces long range communication, it follows that numerical scalability and parallel scalability are often two conflicting objectives. Domain decomposition theory suggests that a good approach for tackling this issue is to augment the DD algorithm with a coarse "grid" problem [ref. 2–4] that is large enough to disseminate significant information globally and yet is small enough to keep computations and communication affordable. Moreover, specialized iterative algorithms are now available for solving efficiently these coarse grid problems on massively parallel processors [ref. 5,6]. Therefore, an ultimate DD based iterative solver is conceivable.

The dual Schur complement method, also known as the Finite Element Tearing and Interconnecting (FETI) method [ref. 7–10], is among the first DD methods to have demonstrated numerical and parallel scalability for the solution of self-adjoint elliptic partial differential equations (PDE) discretized with unstructured finite elements. This method has also been shown to outperform several popular direct and iterative algorithms on both sequential and parallel computing platforms [ref. 1,10]. Essentially, the FETI algorithm can be viewed as a two-step CG-based iterative procedure where subdomain problems with Dirichlet boundary conditions are solved in the preconditioning step, and related subdomain problems with Neumann boundary conditions are solved in the second step. We refer to the FETI method as the dual Schur complement method because on the outset it constructs the dual Schur complement operator. For time-independent elasticity problems, the condition number of the unpreconditioned FETI interface problem grows asymptotically as [ref. 1,11]

$$\kappa \;=\; O\,(\frac{H}{h}) \tag{2}$$

When preconditioned with a subdomain based Dirichlet operator, the condition number of the FETI interface problem varies as [ref. 1,11,12]

$$\kappa \;=\; O\,(1 + log^{\beta}\,(\frac{H}{h}\,)), \quad \beta \leq 3 \tag{3}$$

The conditioning results (2) and (3) highlight the numerical scalability of the FETI method with respect to both the mesh size h and the number of subdomains (which is related to $1/H$). The parallel scalability of this DD method—that is, its ability to achieve larger speedups for larger number of processors—has also been demonstrated on current massively parallel processors for several realistic structural problems [ref. 1,5].

The numerical scalability of the FETI method is due to a coarse problem naturally present in the formulation of the interface problem. In order to guarantee the solvability of the local Neumann problems associated with floating subdomains—that is, subdomains without enough essential boundary conditions to prevent the local stiffness matrices $A^{(s)}$ from being singular—a small auxiliary global problem with at most 6 unknowns per subdomain is solved at each PCG iteration. In [ref. 11], it was shown that this auxiliary problem indeed plays the role of a coarse problem; it provides a satisfactory mechanism for global propagation of the error, which accelerates convergence so that the number of iterations is practically independent of the number of subdomains.

Another numerically scalable algorithm for elasticity problems is the Balancing DD method [ref. 13]. This method is essentially an important improvement of the well-known Neumann-Neumann DD algorithm [ref. 14]. The original Neumann-Neumann DD method can be summarized as a two-step CG-based iterative procedure where subdomain problems with Neumann boundary conditions are solved in the preconditioning step and subdomain problems with Dirichlet boundary conditions are solved in the second step. We refer to this method as the primal Schur complement method because on the outset, it constructs the primal Schur complement operator. The original Neumann-Neumann method lacks a coarse grid problem for propagating the error globally and accelerating convergence. In practice, its rate of convergence deteriorates significantly when more than 8 subdomains are introduced [ref. 15]. As in the FETI method, the coarse problem of the Balancing DD algorithm is defined in terms of the null spaces of the local stiffness matrices. This coarse problem restores the scalability of the original Neumann-Neumann method for a large number of subdomains.

However, it should be noted that the theoretical scalability and optimal conditioning properties of the FETI and Balancing DD methods hold in practice when the subdomains have good and/or comparable aspect ratios, and the partial differential equation to be solved does not feature large (subdomain) coefficient jumps [ref. 1,11]. Each of these two issues represents a different type of subdomain heterogeneity that must be dealt with. In [ref. 16], the authors have proposed a remedy to the first problem in the form of a mesh partitioning optimizer that delivers subdomains with good aspect ratios. In [ref. 17], an ad-hoc scaling procedure was discussed in the context of the Neumann-Neumann DD method for handling potential subdomain heterogeneities. In this paper, we present a rational and superior approach for tackling simultaneously and indifferently all kinds of subdomain heterogeneities. Our methodology is based on energy principles and is best described as a smoothing scheme. However, we also formulate it as a preconditioner. For problems with more than two subdomains, this smoothing scheme generates a coarse grid subproblem that propagates the error globally and accelerates convergence. Because of space limitations, we consider only the case of the FETI or dual Schur complement method. However, the experienced reader will be able to easily transpose the described methodology to the case of the Balancing or primal Schur complement method. We report some preliminary numerical results that demonstrate superior convergence rates for highly heterogeneous elasticity problems.

2. The FETI or dual Schur complement method

The problem to be solved is

$$Ax = b \tag{4}$$

where A is an $n \times n$ symmetric positive semi-definite sparse matrix arising from the finite element discretization of an elasticity problem defined over a region Ω, and b is a right hand side n-long vector representing some prescribed forces. If Ω is partitioned into a set of N_s *disconnected* subdomains $\Omega^{(s)}$, the FETI method consists in replacing Eq. (4) by the equivalent system of subdomain equations

$$\begin{aligned} A^{(s)}x^{(s)} &= b^{(s)} - B^{(s)^T}\lambda \qquad s = 1, \ldots, N_s \\ \sum_{s=1}^{s=N_s} B^{(s)}x^{(s)} &= 0 \end{aligned} \tag{5}$$

where $A^{(s)}$ and $b^{(s)}$ are the restriction of A and b to the disconnected subdomain $\Omega^{(s)}$, λ is a vector of Lagrange multipliers representing the normal derivatives of the primal variable of the problem on the subdomain interface boundary $\Gamma_I^{(s)}$, and $B^{(s)}$ is a signed Boolean matrix which describes the interconnectivity of the subdomains. From a physical viewpoint, the first of Eqs. (5) represents the subdomain equations of equilibrium with Neumann boundary conditions, λ represents the "gluing" forces between the disconnected subdomains (Figure 1), and the second of Eqs. (5) represents the compatibility of the subdomain solutions $x^{(s)}$ across the subdomain interfaces $\Gamma_I = \bigcup_{s=1}^{s=N_s} \Gamma_I^{(s)}$. A more elaborate derivation of Eqs. (5) can be found in [ref. 1,7–11].

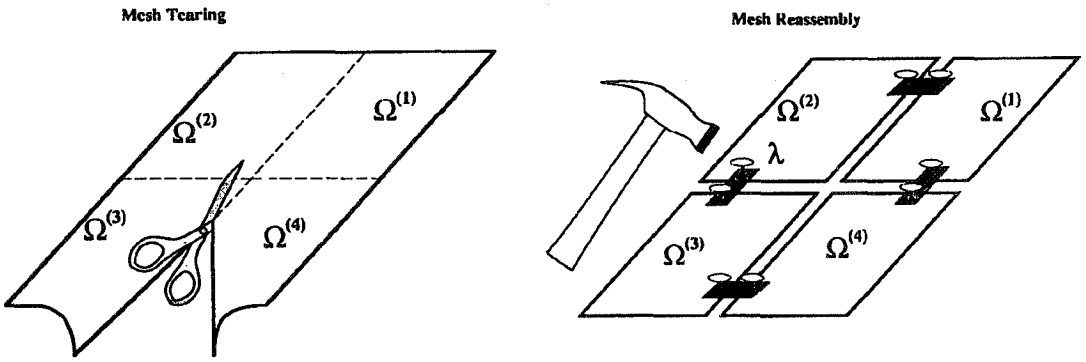


Figure 1. Schematic description of the FETI method.

In general, the mesh partition will contain some N_f floating subdomains, and therefore the Neumann problems

$$A^{(s)}x^{(s)} = b^{(s)} - B^{(s)^T}\lambda \quad s = 1, ..., N_f \tag{6}$$

will be singular. To guarantee the solvability of these problems, we require that

$$(b^{(s)} - B^{(s)^T}\lambda) \perp Ker\,(A^{(s)}) \tag{7}$$

and compute the solution of Eq. (6) as

$$x^{(s)} = A^{(s)^+}(b^{(s)} - B^{(s)^T}\lambda) + R^{(s)}\alpha^{(s)} \tag{8}$$

where $A^{(s)^+}$ is a generalized inverse of $A^{(s)}$ that need not be explicitly computed (see, for example, [ref. 9]), $R^{(s)} = Ker\,(A^{(s)})$ is the null space of $A^{(s)}$, and $\alpha^{(s)}$ is a vector of six or fewer constants (there are, at most, six rigid body modes in a three-dimensional elasticity problem). The introduction of the few additional unknowns $\alpha^{(s)}$ is compensated by the additional equations resulting from (7):

$$R^{(s)^T}(b^{(s)} - B^{(s)^T}\lambda) = 0 \quad s = 1, ..., N_s \tag{9}$$

Substituting Eq. (8) into the second of Eqs. (5) and using Eq. (9) leads (after some algebraic manipulations) to the following FETI interface problem:

$$\begin{bmatrix} F_I & -G_I \\ -G_I{}^T & 0 \end{bmatrix} \begin{bmatrix} \lambda \\ \alpha \end{bmatrix} = \begin{bmatrix} d \\ -e \end{bmatrix} \tag{10}$$

where

$$F_I = \sum_{s=1}^{s=N_s} B^{(s)}A^{(s)^+}B^{(s)^T}; \quad G_I = [B^{(1)}R^{(1)} \quad ... \quad B^{(N_f)}R^{(N_f)}]$$

$$\alpha = [\alpha^{(1)^T} \quad ... \quad \alpha^{(N_f)^T}]^T; \quad d = \sum_{s=1}^{s=N_s} B^{(s)}A^{(s)^+}b^{(s)}; \quad e^{(s)} = b^{(s)^T}R^{(s)}$$

$A^{(s)^+} = A^{(s)^{-1}}$ if $\Omega^{(s)}$ is not a floating subdomain

$A^{(s)^+}$ = a generalized inverse of $A^{(s)}$ if $\Omega^{(s)}$ is a floating subdomain

Clearly, F_I is the sum of independent subdomain operators. Under certain conditions, it can be shown that F_I is the sum of the inverses of the subdomain Schur

complements [ref. 1,11], which justifies the labeling of the FETI method as the dual Schur complement method. It possesses some interesting spectral properties that trigger a superconvergent behavior of a CG algorithm applied to the solution of (10) [ref. 1,11]. Because the above interface problem (10) is *indefinite*, the second step of the FETI method consists in solving it via a preconditioned conjugate *projected* gradient (PCPG) algorithm with a preconditioner $\overline{F_I^{-1}}$ and the projector

$$P = I - G_I \, (G_I{}^T G_I)^{-1} G_I{}^T \tag{11}$$

More specifically, the PCPG FETI algorithm can be formulated as follows [ref. 1]:

1. Initialize

$$\begin{aligned} \lambda^0 &= G_I \, (G_I{}^T G_I)^{-1} e \\ r^0 &= d - F_I \lambda^0 \end{aligned}$$

2. Iterate k = 1, 2, ... until convergence

$$\begin{aligned} \textit{Project} \quad w^{k-1} &= P^T \, r^{k-1} \\ \textit{Precondition} \quad z^{k-1} &= \overline{F_I^{-1}} w^{k-1} \\ \textit{Project} \quad y^{k-1} &= P \, z^{k-1} \\ \zeta^k &= y^{k-1^T} w^{k-1} / y^{k-2^T} w^{k-2} \quad (\zeta^1 = 0) \\ p^k &= y^{k-1} + \zeta^k p^{k-1} \quad (p^1 = y^0) \\ \nu^k &= y^{k-1^T} w^{k-1} / p^{k^T} F_I p^k \\ \lambda^k &= \lambda^{k-1} + \nu^k p^k \\ r^k &= r^{k-1} - \nu^k F_I p^k \end{aligned}$$

(12)

The reader can easily check that, because of the presence of the second projection step, the iterates are independent of the particular choice of the generalized inverse in Eq. (8).

The application of the projection operator P in (12) means that a coarse problem of the form $(G_I{}^T G_I)y = c$ (size $\leq 6 \times N_f \leq 6 \times N_s$) must be solved (twice) in each FETI iteration. It was shown in [ref. 11] that this coarse problem has the expected beneficial effect of coupling all subdomain computations and

propagating the error globally, so that the condition number of the interface problem can be bounded as a function of H/h but is independent of the number of subdomains.

Two preconditioners have been previously developed for the FETI method: (a) a numerically optimal Dirichlet preconditioner that can be written as

$$\overline{F}_I^{D^{-1}} = \sum_{s=1}^{s=N_s} B^{(s)} \begin{bmatrix} 0 & 0 \\ 0 & A_{bb}^{(s)} - A_{ib}^{(s)^T} A_{ii}^{(s)^{-1}} A_{ib}^{(s)} \end{bmatrix} B^{(s)^T} = \sum_{s=1}^{s=N_s} B^{(s)} \begin{bmatrix} 0 & 0 \\ 0 & S_{bb}^{(s)} \end{bmatrix} B^{(s)^T} \tag{13}$$

where $S_{bb}^{(s)}$ denotes the primal Schur complement of subdomain $\Omega^{(s)}$ and the subscripts i and b designate internal and interface boundary unknowns, respectively; and (b) a numerically efficient "lumped" preconditioner that lumps the Dirichlet operator on the subdomain interface unknowns

$$\overline{F}_I^{L^{-1}} = \sum_{s=1}^{s=N_s} B^{(s)} \begin{bmatrix} 0 & 0 \\ 0 & A_{bb}^{(s)} \end{bmatrix} B^{(s)^T} \tag{14}$$

Unlike $\overline{F}_I^{D^{-1}}$, the preconditioner $\overline{F}_I^{L^{-1}}$ is not mathematically optimal. However, it is more economical than $\overline{F}_I^{D^{-1}}$ and has often proved to be more efficient [ref. 1,11].

For practical elasticity problems, the FETI method with either the Dirichlet or Lumped preconditioner is numerically and parallel-wise scalable, when the subdomains have good aspect ratios and no large coefficient jumps. The objective of this paper is to present a third preconditioner that generalizes the two described above and successfully addresses all kinds of heterogeneity problems.

3. Preconditioning with an energy based smoothing procedure

3.1. The two-subdomain problem

In order not to obscure the main idea of this paper by the complexity of the notation needed for a problem with an arbitrary number of subdomains, we consider first the case of a problem with two heterogeneous subdomains. The general case of a system including multiple ($N_s > 2$) and arbitrarily connected subdomains is treated in Section 3.2.

At each iteration of the PCPG FETI algorithm, the matrix vector product $F_I p^k$ produces a jump in the iterate x^k across the subdomain interfaces. (In the sequel, we drop the superscript k for simplicity.) For a heterogeneous problem—for example, a problem with different subdomain stiffnesses—this jump is bound

to be rather large. Elementary mechanics theory suggests that the solution $x^{(s)}$ in the stiffer subdomain $\Omega^{(s)}$ will be closer to the desired converged solution than the solution in the more flexible subdomain. This in turn suggests that the computed solution x should be smoothed after each PCPG iteration as follows:

$$\boxed{\begin{aligned} \tilde{x}_b^{(1)} = \tilde{x}_b^{(2)} = \tilde{x}_I &= (1-a)x_b^{(1)} + a x_b^{(2)} && 0 \le a \le 1 \\ \tilde{x}_i^{(s)} &= A_{ii}^{(s)^{-1}}(b_i^{(s)} - A_{ib}^{(s)}\tilde{x}_b^{(s)}) && \\ &= x_i^{(s)} - A_{ii}^{(s)^{-1}} A_{ib}^{(s)}(\tilde{x}_b^{(s)} - x_b^{(s)}) && s = 1,\ 2 \end{aligned}} \tag{15}$$

Once again, the subscripts i and b designate the internal and interface boundary unknowns. Equations (15) state that first a smoothing of the solution is imposed on the interface boundary between the two subdomains, then a local Dirichlet problem is solved in each subdomain to propagate the beneficial effect of this smoothing to the internal unknowns. Of course, the important question is how to select the optimal smoothing parameter a.

Let δ_I denote the displacement jump on Γ_I defined as

$$\delta_I = x_b^{(2)} - x_b^{(1)} \tag{16}$$

From Eqs. (15) and (16), it follows that $\tilde{x}_b^{(1)}$ and $\tilde{x}_b^{(2)}$ can be rewritten as

$$\tilde{x}_b^{(1)} = x_b^{(1)} + \Delta x_b^{(1)} \qquad \tilde{x}_b^{(2)} = x_b^{(2)} + \Delta x_b^{(2)} \tag{17}$$

where

$$\Delta x_b^{(1)} = a\delta_I \qquad \Delta x_b^{(2)} = -(1-a)\delta_I \tag{18}$$

First, we note that Eqs. (5) can be rewritten as

$$\begin{bmatrix} A_{ii}^{(1)} & A_{ib}^{(1)} & 0 & 0 & 0 \\ A_{ib}^{(1)^T} & A_{bb}^{(1)} & 0 & 0 & B^{(1)^T} \\ 0 & 0 & A_{ii}^{(2)} & A_{ib}^{(2)} & 0 \\ 0 & 0 & A_{ib}^{(2)^T} & A_{bb}^{(2)} & B^{(2)^T} \\ 0 & B^{(1)} & 0 & B^{(2)} & 0 \end{bmatrix} \begin{bmatrix} x_i^{(1)} \\ x_b^{(1)} \\ x_i^{(2)} \\ x_b^{(2)} \\ \lambda \end{bmatrix} = \begin{bmatrix} f_i^{(1)} \\ f_b^{(1)} \\ f_i^{(2)} \\ f_b^{(2)} \\ 0 \end{bmatrix} \tag{19}$$

and that after smoothing, they become

$$\begin{bmatrix} A_{ii}^{(1)} & A_{ib}^{(1)} & 0 \\ A_{ib}^{(1)^T} & A_{bb}^{(1)} + A_{bb}^{(2)} & A_{ib}^{(2)} \\ 0 & A_{ib}^{(2)^T} & A_{ii}^{(2)} \end{bmatrix} \begin{bmatrix} \tilde{x}_i^{(1)} \\ \tilde{x}_I \\ \tilde{x}_i^{(2)} \end{bmatrix} = \begin{bmatrix} b_i^{(1)} \\ b_b^{(1)} + b_b^{(2)} \\ b_i^{(2)} \end{bmatrix} + \begin{bmatrix} 0 \\ r_b \\ 0 \end{bmatrix} \tag{20}$$

where r_b is the interface residual induced by smoothing. From Eq. (19), it follows that

$$r_b = S_{bb}^{(1)} \Delta x_b^{(1)} + S_{bb}^{(2)} \Delta x_b^{(2)} \tag{21}$$

where $S_{bb}^{(s)}$ is the Schur-complement with respect to the interface boundary unknowns of the stiffness matrix of subdomain $\Omega^{(s)}$:

$$S_{bb}^{(s)} = A_{bb}^{(s)} - A_{ib}^{(s)^T} A_{ii}^{(s)^{-1}} A_{ib}^{(s)} \tag{22}$$

Rewriting the induced interface residual in terms of the solution jump δ_I as

$$r_b = r_b(a) = (aS_{bb}^{(1)} + (a-1)S_{bb}^{(2)})\delta_I \tag{23}$$

leads to the conclusion that the optimal parameter a of the smoothing procedure (15) is that which minimizes r_b. However, rather than minimizing directly some norm of r_b, we propose to adopt a Rayleigh-Ritz approach where the smoothed solutions $\tilde{x}^{(1)}(a)$ and $\tilde{x}^{(2)}(a)$ given in Eqs. (15) are viewed as kinematically admissible fields parameterized by a, and to minimize the corresponding energy of the global system. For the two-subdomain problem discussed here, the total energy can be written as

$$\begin{aligned} \mathcal{E}(a) = \frac{1}{2} \begin{bmatrix} \tilde{x}_i^{(1)^T} & \tilde{x}_I^T & \tilde{x}_i^{(2)^T} \end{bmatrix} \begin{bmatrix} A_{ii}^{(1)} & A_{ib}^{(1)} & 0 \\ A_{ib}^{(1)^T} & A_{bb}^{(1)} + A_{bb}^{(2)} & A_{ib}^{(2)} \\ 0 & A_{ib}^{(2)^T} & A_{ii}^{(2)} \end{bmatrix} \begin{bmatrix} \tilde{x}_i^{(1)} \\ \tilde{x}_I \\ \tilde{x}_i^{(2)} \end{bmatrix} \\ - \begin{bmatrix} \tilde{x}_i^{(1)^T} & \tilde{x}_I^T & \tilde{x}_i^{(2)^T} \end{bmatrix} \begin{bmatrix} f_i^{(1)} \\ f_b^{(1)} + f_b^{(2)} \\ f_i^{(2)} \end{bmatrix} \end{aligned} \tag{24}$$

which in view of Eqs. (15)–(23) simplifies to

$$\mathcal{E}(a) = C - 2a\delta_I^T S_{bb}^{(2)} \delta_I + a^2 \delta_I^T (S_{bb}^{(1)} + S_{bb}^{(2)}) \delta_I \tag{25}$$

where C is an expression that does not depend on a. Differentiating $\mathcal{E}$ with respect to a, recalling Eq. (16), and enforcing the condition

$$\frac{d\mathcal{E}}{da} = -2\delta_I^T S_{bb}^{(2)} \delta_I + 2a\delta_I^T (S_{bb}^{(1)} + S_{bb}^{(2)}) \delta_I = 0 \tag{26}$$

finally gives

$$\boxed{\begin{aligned} a^D &= \frac{k^{(2)^D}}{k^{(1)^D} + k^{(2)^D}} \\ k^{(1)^D} &= \delta_I^T S_{bb}^{(1)} \delta_I = (x_b^{(2)} - x_b^{(1)})^T S_{bb}^{(1)} (x_b^{(2)} - x_b^{(1)}) \\ k^{(2)^D} &= \delta_I^T S_{bb}^{(2)} \delta_I = (x_b^{(2)} - x_b^{(1)})^T S_{bb}^{(2)} (x_b^{(2)} - x_b^{(1)}) \end{aligned}} \tag{27}$$

To the authors of this paper, the importance of the above selection of the parameter a is best recognized from a physical viewpoint. Indeed, the smoothing procedure described by Eqs. (15) and (27) consists in treating the two subdomains as two linear springs connected in series, computing the jump of the displacement field at their connection, and redistributing this jump among both springs according to their "relative stiffnesses" $k^{(1)}$ and $k^{(2)}$. While the idea of estimating a local measure of the stiffness of a subdomain to build a scaling matrix for the subdomain preconditioner is not new [ref. 1,17], the derivation of the smoother presented in this paper sheds new light on the precise treatment of all kinds of stiffness heterogeneities. More importantly, Eqs. (27) give for the first time rational estimates $k^{(1)}$ and $k^{(2)}$ of the local measures of the subdomain stiffnesses that clearly contain, among others, the effect of material properties (PDE coefficients), mesh resolutions, and aspect ratios. From a mathematical view point, these constants can be described as the Schur-complement norms of the jump of the solution at the subdomain interfaces. Note that if the two subdomains *and their finite element models* are identical, Eqs. (27) give $k^{(1)} = k^{(2)}$ and $a^D = 1/2$. If the two subdomains differ only in a constant, for example, Young' s modulus E, then Eqs. (27) give $a^D = E^{(2)}/(E^{(1)} + E^{(2)})$. This clearly shows that the smoothing procedure proposed in this paper includes the scaling schemes proposed in [ref. 1,17] as a particular case. *However, if the subdomains do not differ only in one constant, the scaling procedure* $a^D = E^{(2)}/(E^{(1)} + E^{(2)})$ *is not applicable, but the smoothing scheme proposed here is.* Moreover, for problems with more than two subdomains, we will show in Section 3.2 that, unlike the scaling procedure discussed in [ref. 1,17], the smoothing algorithm presented in this paper generates a coarse grid problem that accelerates convergence.

The superscript D in Eqs. (27) is used to highlight the fact that computing the smoothing parameter a^D requires solving subdomain Dirichlet problems that are similar to those induced by the optimal Dirichlet preconditioner (13). Clearly, this establishes that the smoothing procedure (15,27) can be viewed as an improved optimal Dirichlet preconditioner $\overline{\overline{F}}_I^{D^{-1}}$. Alternatively, we can construct

a more economical variant of the proposed smoother where the effect of the interface smoothing is not back-propagated to the subdomain internal unknowns. Following the derivation presented above, the reader can easily check that such a strategy leads to a smoothing procedure similar to that given by Eqs. (15) but where the Schur-complement matrices $S_{bb}^{(s)}$ are replaced by the "lumped" interface stiffness matrices $A_{bb}^{(s)}$, and to the following "lumped" averaging parameter

$$\boxed{\begin{aligned} a^L &= \frac{k^{(2)^L}}{k^{(1)^L} + k^{(2)^L}} \\ k^{(1)^L} &= \delta_I^T A_{bb}^{(1)} \delta_I = (u_b^{(2)} - u_b^{(1)})^T A_{bb}^{(1)} (u_b^{(2)} - u_b^{(1)}) \\ k^{(2)^L} &= \delta_I^T A_{bb}^{(2)} \delta_I = (u_b^{(2)} - u_b^{(1)})^T A_{bb}^{(2)} (u_b^{(2)} - u_b^{(1)}) \end{aligned}} \tag{28}$$

Of course, smoothing with the above lumped strategy can also be viewed as preconditioning with an improved lumped preconditioner $\overline{\overline{F}}_I^{L^{-1}}$. The computational advantages of $\overline{\overline{F}}_I^{L^{-1}}$ are obvious since $A_{bb}^{(s)}$ are readily available, and sparse matrix-vector multiplications rather than forward-backward substitutions are needed to evaluate the smoothing parameter a^L.

3.2. The multiple subdomain problem and the new coarsening operator

Here, we generalize the smoothing procedure presented in the previous section to the case of multiple ($N_s > 2$) and arbitrarily connected subdomains.

Let $b^{(s)}$ denote the restriction of the Boolean operator $B^{(s)}$ defined in Eqs. (5) to the interface boundary $\Gamma_I^{(s)}$ of a given subdomain $\Omega^{(s)}$. Using the internal/interface subdomain partitioning of the unknowns we have

$$B^{(s)} = [0 \quad b^{(s)}] \tag{29}$$

The interface boundary of each subdomain can be broken into edges and therefore $b^{(s)}$ can be partitioned as

$$b^{(s)} = [b^{(s),i} \quad b^{(s),j} \quad \dots \quad b^{(s),l}] \tag{30}$$

where $b^{(s),j}$ is the restriction of $b^{(s)}$ to the j-th edge of $\Gamma_I^{(s)}$. Note that Eq. (30) implies that every interface point is assigned to one and only one edge, and therefore *a crosspoint is treated as a single point edge.* Finally, we introduce the *unsigned* equivalents of $B^{(s)}$, $b^{(s)}$, and $b^{(s),j}$ and designate them with a circumflex.

Using this notation, the jump of the solution x across an edge j between two subdomains $\Omega^{(s)}$ and $\Omega^{(q)}$ can be written as

$$\delta_I^{(s,t),j} \;=\; \hat{b}^{(s),j}\, x_b^{(s),j} - \hat{b}^{(t),j}\, x_b^{(t),j} \;=\; x_b^{(s),j} - x_b^{(t),j} \tag{31}$$

where $x_b^{(s),j}$ is the trace of the subdomain solution $x^{(s)}$ on the the edge j. Consequently, the generalization to an arbitrary number of subdomains of the smoothing procedure proposed in Eqs. (15) is given by

$$\boxed{\begin{aligned} \tilde{x}_b^{(s),j} &= \sum_{\Gamma_I^{(t)} \ni j} \beta^{(t),j}\, \hat{b}^{(s),j^T} \hat{b}^{(t),j}\, x_b^{(t),j} \\ \tilde{x}_i^{(s)} &= -A_{ii}^{(s)^{-1}} A_{ib}^{(s)} \Delta x_b^{(s)} + x_i^{(s)} \end{aligned}} \tag{32}$$

It remains to find the optimal values of the edge coefficients $\beta^{(s),j}$. For this purpose, we follow conceptually the same Rayleigh-Ritz approach presented in Section 3.1. Let $\Delta x_b^{(s),j}$ be defined as

$$\Delta x_b^{(s),j} \;=\; \tilde{x}_b^{(s),j} - x_b^{(s),j} \tag{33}$$

If the coefficients $\beta^{(s),j}$ are constrained to have a unit sum

$$\boxed{\sum_{\Gamma_I^{(s)} \ni j} \beta^{(s),j} \;=\; 1} \tag{34}$$

then from Eqs. (31)–(33) it follows that

$$\Delta x_b^{(s),j} \;=\; -\hat{b}^{(s),j^T} \sum_{\Gamma_I^{(t)} \ni j,\; t \neq s} \beta^{(t),j}\, \delta_I^{(s,t),j} \tag{35}$$

For a problem with an arbitrary number of subdomains, the total energy can be written as

$$\mathcal{E}(\beta^{(s),j}) \;=\; \sum_{s=1}^{s=N_s} \tilde{u}^{(s)^T} A^{(s)} \tilde{u}^{(s)} - \tilde{u}^{(s)^T} f^{(s)} \tag{36}$$

If the effect of interface smoothing is back-propagated to the subdomain interiors (Dirichlet smoothing), $\mathcal{E}(\beta^{(s),j})$ can be rewritten as

$$\begin{aligned} \mathcal{E}(\beta^{(s),j}) &= \sum_{s=1}^{s=N_s} \tilde{x}_b^{(s)^T} S_{bb}^{(s)} \tilde{x}_b^{(s)} - \tilde{x}_b^{(s)^T} (f_b^{(s)} - A_{ib}^{(s)} A_{ii}^{(s)^{-1}} f_i^{(s)}) \\ &= C + \frac{1}{2} \sum_{s=1}^{s=N_s} \Delta x_b^{(s)^T} S_{bb}^{(s)} \Delta x_b^{(s)} \end{aligned} \tag{37}$$

where C is a function that does not depend on the edge parameters $\beta^{(s),j}$. On the other hand, if the effect of interface smoothing is not back-propagated to the subdomain interiors (lumped smoothing), $\mathcal{E}(\beta^{(s),j})$ will have an expression similar to that of (37) but with $A_{bb}^{(s)}$ replacing every occurrence of $S_{bb}^{(s)}$. Minimizing the energy with respect to the edge smoothing parameters ($\frac{\partial \mathcal{E}}{\partial \beta^{(s),j}} = 0$) leads after some algebraic transformations to

$$\boxed{\sum_{\substack{\Gamma_I^{(s)} \ni j \\ s \neq q}} \sum_{\Gamma_I^{(k)} \ni s} \sum_{\substack{\Gamma_I^{(p)} \ni k \\ p \neq s}} \beta^{(p),k} (\delta_I^{(q,s),j}\ \hat{b}^{(s),j^T} [S_{bb}^{(s)}]_{j,k}\ \hat{b}^{(s),k} \delta_I^{(p,s),k}) = 0 \quad \forall j,\ \Gamma^{(q)} \ni j} \tag{38}$$

where $[S_{bb}^{(s)}]_j$ is the Schur-complement of $A^{(s)}$ associated with the edge j and k. Hence, the edge smoothing parameters $\beta^{(s),j}$ are given by the solution of a coarse auxiliary problem of size as small as the number of edges in the mesh partition. There is no question that the above system of equations (38) is quite complicated to read. However, there is also no question that it is easy to program since the $\hat{b}^{(s),k}$ are Boolean operators.

3.3. Dealing with the variable preconditioner

Since the values of the jumps of the iterate x^k across the subdomain edges change in each iteration k, it follows from Eqs. (27) and (28) that the proposed preconditioner changes in every FETI PCPG iteration. Of course, one can always freeze the coefficients $\beta^{(s),j}$ after the first or a few iterations. However, a reorthogonalization is always used in practice with the FETI method [ref. 1], so that the variation of the preconditioner with the iteration number is not an issue. We note that we have previously demonstrated (see [ref. 1], for example) that this reorthogonalization is cost-efficient because it is applied only to the interface problem, and it does not significantly increase the total CPU time for solving the global problem.

4. Numerical results

In order to demonstrate the potential of the proposed preconditioner, we consider the plane stress analysis of a two-dimensional heterogeneous structure comprising steel and rubber subcomponents (see Figure 2). The global structure is clamped at one end and is subjected to a horizontal and vertical point loads at the top of the other end. The nearly incompressible rubber subcomponents are characterized by a Young modulus $E^{(rubber)} = 5.0 \times 10^7\ N/mm^2$ and a Poisson ratio $\nu^{(rubber)} = 0.48$, and the steel subcomponents by $E^{steel} = 2.05 \times 10^{11}\ N/mm^2$

and $\nu^{steel} = 0.3$. The numerical difficulties of this problem are spurred by its high degree of heterogeneity, measured here by the ratio $E^{steel}/E^{rubber} = 4098$, and by the presence of a crosspoint between extremely stiff and extremely flexible subdomains.

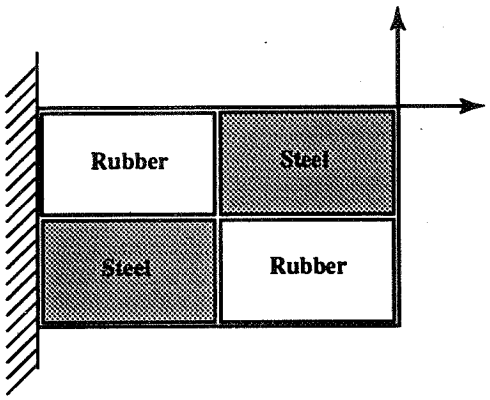


Figure 2. A heterogeneous steel/rubber plane elasticity problem.

Four different meshes are constructed for the solution of this problem using 4, 16, 64, and 256 subdomains. All meshes verify $H/h = 8$. The FETI method is used with: (a) the Dirichlet preconditioner weighted by the number of subdomains connected to an interface point (DR) and (b) the smoothing based new Dirichlet preconditioner summarized in Eqs. (32) and (38) (SMTH). The convergence results are reported in Table I where N_{eq} and N_{itr} denote, respectively, the number of equations associated with each finite element discretization and the number of iterations. All computations are performed using MATLAB.

Table I. Solution of a Steel/Rubber Heterogeneous Plane Elasticity Problem

FETI solver
Dirichlet precond. (DR) vs. new smoothing based Dirichlet precond. (SMTH)
Global convergence criterion: $||Ax - b||_2 \leq 10^{-6} \times ||b||_2$

H	h	N_{eq}	N_s	N_{itr} (DR)	N_{itr} (SMTH)
1/2	1/16	612	4	35	11
1/4	1/32	2520	16	105	39
1/8	1/64	10224	64	153	80
1/16	1/128	41478	256	246	82

The FETI method is shown to converge three times faster with the new smoothing based Dirichlet preconditioner than with the original Dirichlet preconditioner.

References

1. Farhat, C.; and Roux, F. X.: Implicit parallel processing in structural mechanics, Computational Mechanics Advances, vol. 2, 1994, pp. 1-124.

2. Axelsson, O.; and Gustafsson, I: Preconditioning and two-level multigrid methods of arbitrary degree of approximation, Math. Comp., vol. 40, 1983, pp. 219-242.

3. Bramble, J. H.; Pasciak, J. E. ; and Schatz A. H.: The construction of preconditioners for elliptic problems by substructuring, I, Math. Comp., vol. 47, 1986, pp. 103-134.

4. Kočvara, M.; and Mandel, J,: A multigrid method for three-dimensional elasticity and algebraic convergence estimates, Appl. Math. Comput., vol. 23, 1987, pp. 121-135.

5. Farhat, C.; and Chen, P. S.: Tailoring domain decomposition methods for efficient parallel coarse grid solution and for systems with many right hand sides, Contemporary Mathematics, vol. 180, 1994, pp. 401-406.

6. Farhat, C.; Crivelli, L.; and Roux, F. X.: Extending substructure based iterative solvers to multiple load and repeated analyses, Comput. Meths. Appl. Mech. Engrg., vol. 117, 1994, pp. 195-209.

7. Farhat, C.: A Lagrange multiplier based divide and conquer finite element algorithm, J. Comput. Sys. Engrg., vol. 2, 1991, pp. 149-156.

8. Farhat, C: A saddle-point principle domain decomposition method for the solution of solid mechanics problems, in: D. E. Keyes, T. F. Chan, G. A. Meurant, J. S. Scroggs and R. G. Voigt, ed., Proc. Fifth SIAM Conference on Domain Decomposition Methods for Partial Differential Equations, SIAM, 1991, pp. 271-292.

9. Farhat, C; and Roux, F. X.: A method of finite element tearing and interconnecting and its parallel solution algorithm, Internat. J. Numer. Meths. Engrg., vol. 32, 1991, pp. 1205-1227.

10. Farhat, C.; and Roux, F. X.: An unconventional domain decomposition method for an efficient parallel solution of large-scale finite element systems, SIAM J. Sc. Stat. Comp., vol. 13, 1992, pp. 379-396.

11. Farhat, C.; Mandel, J.; and Roux, F. X.: Optimal convergence properties of the FETI domain decomposition method, Comput. Meths. Appl. Mech. Engrg., vol. 115, 1994, pp. 367-388.

12. Mandel, J.; and Tezaur, R.: Convergence of a substructuring method with Lagrange multipliers, Copper Mountain Conference on Multigrid Methods, NASA CP-3339 , 1996.

13. Mandel, J.: Balancing domain decomposition, Comm. Appl. Num. Meth., vol. 9, 1993, pp. 233-241.

14. Bjorstad, P. E.; and Widlund O. B.: Iterative methods for solving elliptic problems on regions partitioned into substructures, SIAM J. of Num. Anal., vol. 23, 1986, pp. 1097-1120.

15. LeTallec, P.; De Roeck, Y. H.; and Vidrascu, M.: Domain decomposition methods for large linearly elliptic three dimensional problems, J. Comp. Appl. Math., vol. 34, 1991, pp. 93-117.

16. Farhat, C.; Maman, N.; and Brown, G.: Mesh partitioning for implicit computations via iterative domain decomposition: impact and optimization of the subdomain aspect ratio, Internat. J. Numer. Meths. Engrg., vol. 38, 1995, pp. 989-1000.

17. LeTallec, P.: Domain decomposition methods in computational mechanics, Computational Mechanics Advances, vol. 1, 1994, pp. 121-220.

18. P. Morice, Transonic computations by a multidomain technique with potential and Euler solvers, in: J. Zienep and H. Ortel, eds., Symposium Transsonicum III, IUTAM Symposium, 1989.

HIGH PERFORMANCE PARALLEL MULTIGRID ALGORITHMS FOR UNSTRUCTURED GRIDS

Paul O. Frederickson

Math Cube Associates, Inc.
PO Box 66, Los Alamos, NM 87544

pof@mathcube.com

SUMMARY

We describe a high performance parallel multigrid algorithm for a rather general class of unstructured grid problems in two and three dimensions. The algorithm PUMG, for parallel unstructured multigrid, is related in structure to the parallel multigrid algorithm PSMG introduced by McBryan and Frederickson, for they both obtain a higher convergence rate through the use of multiple coarse grids. Another reason for the high convergence rate of PUMG is its smoother, an approximate inverse developed by Baumgardner and Frederickson.

INTRODUCTION

The fundamental task of the algorithm PUMG is to solve a large sparse linear system of the form

$$Au = v \tag{1}$$

as efficiently as possible, since it will likely need to solve it repeatedly. We assume that a tolerance ϵ has been given, and that an approximate solution u is acceptable if the residual

$$r = v - Au \tag{2}$$

satisfies $\| r \| < \epsilon$. For clarity we refer to u as an $\epsilon-approximate\ \ solution$ when this is the case. In many cases the sparse matrix A will be symmetric and positive definite, which makes the theoretical analysis easier, but we observe excellent convergence for rather general nonsymmetric systems as well. We assume that eqn.(1) is the discretization, by some linear process, of the continuous linear system

$$\mathcal{A}u = v \tag{3}$$

on a smooth two- or three-dimensional manifold Ω. One of the advantages of the algorithm PUMG is that this may as well be an unstructured discretization. Internally, PUMG uses a *cell-based*

discretization algorithm to construct the coarse grid approximations, even though the given sparse linear system (1) may have been the result of a quite different discretization process.

The continuous linear system (2) may well be the variational equation of a nonlinear system which is to be solved by Newton's method, or it may represent an implicit time step in the evolution of a hyperbolic system

$$\mathcal{F}(t, u(x,t)) \;=\; v(x,t). \tag{4}$$

PUMG was developed for the implicit time step of a hyperbolic system of this form, namely the shallow water equations on a sphere, which explains our interest in efficiency.

Higher order interpolation is the first key to higher performance in the unstructured multigrid algorithm PUMG. Since the concept of *polynomial reconstruction* on which we base our interpolation is not yet widely known, we devote the third section to a clarification of this idea and a description of how it is used to construct the interpolation operator Q used in PUMG.

The second key to higher performance in PUMG is the use of more than one coarse grid at every level, in a manner somewhat analogous to that in the algorithm PSMG [13][14]. We make this concept of *tree structured multigrid* more precise in the fourth section. The third key is the use of a well tailored *local approximate inverse* in the smoothing step of PUMG. In the fifth section we discuss the *quadrature based smoother* **QBS** introduced by Baumgardner and Frederickson at the 1993 Copper Mountain Conference and contrast it with the **ILU**, *LS* and *DB* smoothers.

UNSTRUCTURED CELLULAR DECOMPOSITIONS

There is no longer any doubt about the advantages of unstructured grids in the high precision solution of many real world problems. Their flexibility allows the gridding of complicated domain shapes more readily, and allows local mesh refinement in regions where the solution develops high gradients. The early work of Bank and Sherman [3–5], Bank and Rose [2], and others gives ample evidence of this, along with the fact that multigrid can be adapted to the solution of these problems. Convergence rates for unstructured multigrid algorithms remain somewhat slower than they are for classic multigrid however, which is one of the motivations for the current algorithm. We observe that general cellular decompositions of a domain offer computational advantages over decompositions that use only tetrahedra and hexahedra. For example, several times as many tetrahedra of a given maximal diameter are required to fill a region as are required for well-proportioned cells. This increases the cost of most aspects of the computation. We claim that the concept of cell center is not important in cellular discretizations, as it is better to think of a quantity as distributed over the cell rather than located at any one point. For higher order accuracy this distribution will, of course, be nonuniform.

HIGHER ORDER POLYNOMIAL RECONSTRUCTION

The first key to high performance in a multilevel solver for unstructured grids is a high-order interpolation operator Q for transferring a subgrid solution to the next higher level. From the cellular discretization viewpoint, this implies a model for the distribution within each coarse cell of the variable to be interpolated to the finer cells.

We will construct this distribution using *a polynomial reconstruction* algorithm R that constructs a polynomial p_i in each cell C_i using the state u_j in neighboring cells C_j. Exactly how we choose a neighborhood will depend on several factors, including the desired degree k of the reconstruction. We will require at least $\binom{k}{d}$ cells, including cell C_i itself, to reconstruct a polynomial of degree k in d dimensions. When the cellular decomposition is fairly uniform we usually find that the cells contiguous to a given cell, together with that cell, form a sufficiently large neighborhood to support quadratic reconstruction. Boundary cells will need to use a more one sided neighborhood if they require the same degree of reconstruction. The effect of this is not so severe if there is a layer or two of smaller cells near the boundary, with further refinement in the corners. This boundary refinement is often advantageous for a variety of other reasons, one of the advantages of an unstructured grid.

To make the concept of neighborhood precise we denote by N_i the set of indices j of the cells C_j in the chosen neighborhood of cell C_i. For simplicity we will assume that the system we are solving is scalar, and we will represent it with a *state vector* $u = \langle u_i \rangle$. The vector of polynomials that results from the reconstruction will be denoted $p_i = \langle p_i \rangle$ in the following discussion. In each case the index i runs over the list of cells in the unstructured grid.

We will define an operator R that constructs a polynomial of degree k in each cell to be a *k-exact polynomial reconstruction* operator if it satisfies the following three axioms:

Axiom 1: The operator R preserves cell averages. If we denote by S the discretization operator that computes the average of a variable over each cell, then R satisfies

$$p = Ru \implies u = Sp.$$

Axiom 2: The operator R is *k-exact* in that it reproduces polynomials of degree k exactly:

$$p \in \mathcal{P}^k, \; u = Sp \implies p = Ru.$$

Axiom 3: The operator R is *local* in that it constructs the polynomial in cell C_i using the values of u_j in neighboring cells C_j only:

$$u_j = 0, \; j \in \mathcal{N}_i \implies (Ru)_i = 0.$$

Note that Axiom 1 simply states that R is a right inverse of the averaging operator S:

$$SR = I \tag{5}$$

and Axiom 2 states that R is a left inverse to S restricted to the space of degree k polynomials:

$$RSp = p, \; p \in \mathcal{P}^k. \tag{6}$$

From these two equations it should not be surprising that the reconstruction operator R is a pseudo-inverse of some sort. In fact, the construction below builds R as a sparse block matrix, each row of which constitutes a Moore-Penrose pseudo-inverse of S.

This concept of k-exact reconstruction on unstructured grids was introduced by Frederickson and Barth [12] for use in a high-order CFD solver on unstructured grids, and has been further developed by Barth [6], Coirier and Powell [9], and others for a variety of applications in fluid dynamics. This appears to be the first application or k-exact reconstruction to an unstructured and non-nested multigrid solver.

Numerical Construction of *R*

We prescribe a unique operator R satisfying these three axioms by choosing the one with the smallest coefficients, in the least squares sense. This description of R as the solution of a variational problem allows it to be constructed with the linear least squares procedure that we describe below. Better yet, this construction is fairly inexpensive, as each row of R, when represented as a block matrix, is constructed independently. The computation proceeds as follows:

Step 1: For each j, choose a list N_j of n_j cells that neighbor the cell C_j, (including the cell C_j itself), enough so that $n_j \geq \binom{k}{d}$.

Step 2: For each j, choose a local basis $< p_1, p_2, \ldots, p_m >$ for $\mathcal{P}^k$ on the cell C_j, compute the averages of these $m = \binom{k}{d}$ basis functions on cell C_j, and enter these as the j^{th} column of the matrix W:

$$W_{i,j} = (Sp_i)_j.$$

Step 3: For each j, form the m by n_j matrix V by deleting all columns of W not in N_j and translating the remaining columns to the local coordinate system of cell C_j. Beginning with a QR-factorization of the matrix V^H, compute the Moore-Penrose pseudo-inverse of V and enter this as the j^{th} row of the sparse block matrix representation of the reconstruction operator R. During the QR-factorization make sure that the matrix is of full rank, otherwise the list of neighbors must be increased.

In practice we avoid forming the matrix V^H explicitly for input to the QR algorithm, and instead apply the *transpose* of the QR-factorization algorithm to V itself. This modified algorithm factors V into the product of a lower triangular matrix L and an orthogonal matrix Q, and therefore is sometimes referred to as the LQ-factorization of a matrix. We recommend the use of either Householder rotations or Givens reflections in carrying out the factorization. Finally, we wish to warn the reader of a notational conflict: the Q and R of QR-factorization have nothing to do with the reconstruction operator R or the interpolation operator Q that we discuss next.

The Operator *Q*

The interpolation operator Q that transfers the state u from a coarse grid to a finer grid is constructed from the reconstruction operator R in such a way that each application of the operator Q is equivalent to the following three steps:

Step 1: Reconstruct $p_i = (Ru)_i$ on each cell C_i of the coarse grid.

Step 2: Intersect each cell C_i of the coarse grid with every cell $C_{i'}$ of the finer grid, and transfer p_i to that intersection.

Step 3: On each cell $C_{i'}$ of the fine grid apply the averaging operator S to the resulting piecewise polynomial function.

For the sake of computational efficiency, however, this use of the reconstruction operator R in constructing Q is carried out only during the setup phase, and results in an explicit sparse matrix representation of the interpolation operator Q. The most difficult step in this construction is forming the grid which is the intersection of the coarse grid and the fine grid because our grids are not generally nested. In saying this we assume that the averaging operator S, which is able to compute the average over an arbitrary cell of a polynomial $p \in \mathcal{P}^k$, is already available.

TREE STRUCTURED PARALLEL MULTIGRID

The second key to rapid convergence is the use of more than one coarse grid at every level, resulting in a tree structured algorithm. The convergence is sufficiently faster for difficult problems to justify the somewhat greater computational complexity, which is not excessive if the subgrids are coarse enough. The code complexity is not significantly greater, as the code fragment shown in Figure 1 demonstrates. The variable `node` in this routine points to a data structure that contains everything that would be needed at one level of an ordinary multigrid algorithm such as FAPIN; in addition, this data structure includes pointers to the nodes that contain the next finer grid and pointers to the nodes that contain all coarser grids.

LOCAL APPROXIMATE INVERSES AS SMOOTHERS

The third key to the strong convergence of PUMG is the use of a well engineered *local approximate inverse* Z to remove the high frequency part of the error via the two step smoother

$$r \leftarrow v - Au \tag{7}$$

$$u \leftarrow u + Zr. \tag{8}$$

The widely used **ILU** smoother, or *incomplete Cholesky* smoother introduced by Van der Vorst [10] and Meijerink and Van der Vorst [15], is almost of this type, but not quite, for although $Z = U^{-1}bL^{-1}$ is implicitly local and can be applied at much the same cost as the other three smoothers described below, it has a derivation which differs considerably. The idea behind incomplete Cholesky is to follow the Cholesky algorithm for computing the lower triangular factor $\mathbf{L}$ and the upper triangular factor $\mathbf{U}$ with one exception: as each element is computed, it is set equal to zero if it is located outside

```
void PUMG::solve( pumg_node *node ){
    if( NULL == node->upper_node ){
       node->data_in_u = 1;
       resid( node->r, node->v, node->A, node->u, node );
    }
    else{
       node->data_in_u = 0;
       project( node->v, node->P, node->upper_node->r, node );
       copy( node->r, node->v, node );
    }
    smooth( node->u, node->Z, node->r, node );
    resid( node->r, node->v, node->A, node->u, node );
    for( int i=0; i<node->num_sub_grids; i++ ){
       PUMG::solve( &(node->lower_node[i][0]) );
    }
    if( 0 < node->num_sub_grids ){
       resid( node->r, node->v, node->A, node->u, node );
    }
    smooth( node->u, node->Z, node->r, node );
    if( NULL != node->upper_node )
       interp( node->upper_node->u, node->Q, node->u, node );
    }
```

Figure 1. The main loop of PUMG.

of the prescribed neighborhood N of the identity. In the earliest version this was taken to be exactly the nonzero set of the sparse matrix A, and in later versions this was enlarged for difficult problems, to avoid zeroing our elements of significant size.

The ultimate goal of the approximate inverse smoother Z is to minimize the spectral radius of the whole multigrid cycle, with the sparsity pattern of Z as the only constraint. Although it is easy enough to construct such an optimal Z for constant coefficient periodic problems, as demonstrated in [14], it becomes rather expensive for general unstructured grids. A much less costly approach is to focus on the smoother step alone, and construct Z so that

$$\|(I - AZ)r\| \tag{9}$$

is small for all r of high spatial frequency. More precisely, we would like to minimize the maximum of this expression as r varies over the null space of the projection operator P. Alternatively, one could focus on the errors rather than the residuals, and seek Z such that

$$\|(I - ZA)e\| \tag{10}$$

is small for all e such that Ae is in the same null space. These are global optimization problems, however, and expensive enough that we would prefer a local alternative. We describe three effective alternatives below.

The *LS* Approximate Inverse

The first of these, the *LS* approximate inverse Z, is constructed explicitly as the minimum of the quadratic functional

$$M(Z) = \|I - AZ\|_{\mathcal{F}}^2 = \sum_{i,j} \left|(I - AZ)_{i,j}\right|^2 = \sum_i \left(\left(I - Z^H A^H\right)(I - AZ)\right)_{i,i} \tag{11}$$

(the square of the Frobenius norm) subject to the constraint that Z must vanish outside chosen neighborhood $\mathcal{N}$ of the identity. The quadratic functional M has the property that the minimizing Z is easily computed one column at a time. To see this, let z^k denote the k^{th} column of Z, let $y^k = Ax^k$, and let N^k denote the k^{th} column of N, namely the set of indices i such that z_i^k (or $Z_{i,k}$) is allowed to be non-zero. Then

$$M(Z) = \sum_k M^k(Z), \tag{12}$$

where

$$M^k(Z) = \sum_i \left|\left(\delta_{i,k} - y_i^k\right)\right|^2. \tag{13}$$

Thus to construct the optimal Z we only need to choose z^k to minimize $M^k(Z)$ and do so for each k. But this optimal z^k satisfies the system of equations

$$\sum_j B_{i,j} z_j^k = A_{i,k}^H, \quad i \in N^k, \tag{14}$$

where

$$B_{i,j} = \begin{cases} \left(A^H A\right)_{i,j} & \text{if } i \in N^k \text{ and } j \in N^k \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

If we wish to focus on the errors, rather than the residuals, and ask that Z minimize $\|I - ZA\|_{\mathcal{F}}^2$ subject to the same sparsity constraints, we find that the rows of Z satisfy essentially the same equation, but with AA^H replacing $A^H A$ in the definition of the small matrix B, and with A rather than A^H on the right hand side.

The *LS* approximate inverse was introduced by M. W. Benson in his 1973 thesis, referenced in [7], and has served as an effective smoother in the v-cycle multigrid algorithm FAPIN for twenty years now. (See [8] and [1] and the reports referenced therein.)

The *DB* Approximate Inverse

When the sparse matrix A is self adjoint there is another approximate inverse that is even easier to compute because it doesn't involve forming blocks of the normal matrix $A^H A$; in most situations this approximate inverse works almost as well as the *LS* approximate inverse in a defect-correction smoother. The idea is to weight the quadratic functional *M(Z)* with A^{-1}:

$$M(Z) = \sum_i \left(\left(I - Z^H A^H\right) A^{-1}(I - AZ)\right)_{i,i} = \sum_i \left(I - Z^H A - AZ + Z^H AZ\right)_{i,i}, \tag{16}$$

and observe that the columns of the optimizing Z now satisfy the system of equations

$$\sum_j A_{i,j} z_j^k = \delta_{i,k}, \quad i \in N^k. \tag{17}$$

This is also discussed in [7] and the reports listed therein.

The Quadrature Based Smoother (QBS)

For an even more effective smoother we recommend the *quadrature based smoother* introduced by Baumgardner and Frederickson at the 1993 Copper Mountain conference. The idea of behind QBS is to minimize a quadratic functional of the form

$$\mathcal{F}(\mathbf{Z}) = \sum_i c_i \|(\mathbf{I} - \mathbf{ZA})\mathbf{e}_i\|_{\mathbf{A}}^2, \tag{18}$$

where the carefully chosen set of errors $\mathbf{e}_i$, together with the weight c_i associated with each, are chosen to represent the expected error before smoothing. In particular, they are chosen so that the associated residuals $\mathbf{r}_i = \mathbf{A}\mathbf{e}_i$ span the null space of the projection operator $\mathbf{P}$. This quadratic functional may be viewed as quadrature approximation to an energy integral of the form

$$\mathcal{F}(\mathbf{Z}) = \int_{\mathcal{L}_2} \|(\mathbf{I} - \mathbf{ZA})\mathbf{e}\|_{\mathbf{A}}^2 d\mu(\mathbf{e}),$$

where the measure μ is chosen to represent the energy in the error just before smoothing. Indeed, they may be put exactly into this integral form by the simple expedient of choosing the measure μ to vanish except at the points $\mathbf{e}_i$ and giving it the value c_i there. It is this viewpoint that allows us to refer to the errors $\mathbf{e}_i$ as quadrature points in $\mathcal{L}_2$.

Because these quadratic functionals are positive definite, they each have a unique minimum when restricted to the space of sparse matrices $\mathbf{Z}$ of given sparsity structure. In every case the algorithm for constructing this optimum $\mathbf{Z}$ is entirely local, because the first variation $\delta\mathcal{F}$ is a block diagonal matrix, each block of which involves only one row of $\mathbf{Z}$ and nearby rows of $\mathbf{A}$. To be specific, the (sparse) row $\mathbf{z} = \mathbf{z}_i$ of the sparse matrix $\mathbf{Z}$ satisfies an equation of the form

$$W\mathbf{z} = b, \tag{19}$$

in which W and b are constructed as follows. For each of the $\mathbf{e}_k$, evaluate $\mathbf{y}_k = s_k\mathbf{A}\mathbf{e}_k$, where $\mathbf{s}_k$ is the sparsity of the row $\mathbf{e}_k$. Let e_k denote the element of the vector $\mathbf{e}_k$ in that row. Then

$$W = \sum_k c_k \mathbf{y}_k \mathbf{y}^T{}_k \quad \text{and} \quad b = \sum_k c_k \mathbf{y}_k e_k. \tag{20}$$

Because Z is sparse this system is easily solved, for the order of the system is the number of nonzeros in a row of Z. We have found that by choosing the quadrature points e_k and the weights c_k appropriately we are able to construct a nearly optimum smoother. In very difficult situations, with strongly varying coefficients and/or cell sizes in some localities, we allow Z to fill a larger neighborhood of the identity in these localities in order to obtain a spatially uniform rate of convergence. All in all, our best smoother for these problems is QBS. When there are few enough of e_k, this least squares approach reduces to an exact solve, and the resulting smoother anihilates these errors exactly. In this case we might also refer to Z as a *collocation approximate inverse* by analogy with other collocation algorithms in numerical analysis.

CONCLUSION

We find that higher order interpolation, important to faster convergence in an unstructured multigrid algorithm, can be effectively constructed using the polynomial reconstruction algorithm of Barth and Frederickson. Tree structured multigrid algorithms are an additional means of gaining faster convergence in unstructured multigrid. The most efficient smoothers for unstructured problems of this sort are the quadrature based smoothers indtroduced in 1993 by Baumgardner and Frederickson.

We would like to conclude with a note of thanks to Craig and Marietta Douglas for the bibliography [11] which they have constructed for our use, and to Steve McCormick for organizing this sequence of conferences. The catalytic effect on multigrid research of both of these efforts is clear to all of us.

REFERENCES

[1] R. N. Banerjee and M. W. Benson. An approximate inverse based multigrid approach to the biharmonic problem. *Int. J. Comput. Math.*, 40:201–210, 1991.

[2] R. E. Bank and D. J. Rose. Analysis of a multilevel iterative method for nonlinear finite element equations. *Math. Comp.*, 39:453–465, 1982.

[3] R. E. Bank and A. H. Sherman. Algorithmic aspects of the multi–level solution of finite element equations. In I. S. Duff and G. W. Stewart, editors, *Sparse Matrix Proccedings 1978*, pages 62–89, Philadelphia, 1979. SIAM.

[4] R. E. Bank and A. H. Sherman. A multi–level iterative method for solving finite element equations. In I. S. Duff and G. W. Stewart, editors, *Proceedings of the Fifth Symposium on Reservoir Simulation*, pages 117–126, Dallas, 1979. Society of Petroleum Engineers of AIME.

[5] R. E. Bank and A. H. Sherman. The use of adaptive grid refinement for badly behaved elliptic partial differential equations. *Math. Comput. Simulation*, 22:18–24, 1980.

[6] T. J. Barth. Recent developments in high order k-exact reconstruction on unstructured meshes. AIAA paper no. 93-0668, 1993.

[7] M. W. Benson and P. O. Frederickson. Iterative solution of large sparse linear systems arising in certain multidimensional approximation problems. *Utilitas Mathematica*, 22:127–140, 1982.

[8] M. W. Benson and P. O. Frederickson. Fast pseudo–inverse algorithms on hypercubes. In S. F. McCormick, editor, *Multigrid Methods: Theory, Applications, and Supercomputing*, volume 110 of *Lecture Notes in Pure and Applied Mathematics*, pages 23–33. Marcel Dekker, New York, 1988.

[9] W. J. Coirier and K. G. Powell. An accuracy assessment of Cartesian-mesh approaches for the Euler equations. AIAA paper no. 93-3335, 1993.

[10]H. A. Van der Vorst. High performance preconditioning. *SIAM J. Sci. Stat. Comp.*, 10:1175–1184, 1989.

[11]C. C. Douglas and M. B. Douglas. MGNet Bibliography. In mgnet/bib/mgnet.bib, on anonymous ftp server casper.cs.yale.edu, Yale University, Department of Computer Science, New Haven, CT.

[12]P. O. Frederickson and T. J. Barth. Higher order solution of the Euler equations on unstructured grids using quadratic reconstruction. AIAA paper no. 90-0013, 1990.

[13]P. O. Frederickson and O. A. McBryan. Parallel superconvergent multigrid. In S. F. McCormick, editor, *Multigrid Methods: Theory, Applications, and Supercomputing*, volume 110 of *Lecture Notes in Pure and Applied Mathematics*, pages 195–210. Marcel Dekker, New York, 1988.

[14]P. O. Frederickson and O. A. McBryan. Normalized convergence rates for the {psmg} method. *SIAM J. Sci. Stat. Comput.*, 12:221–229, 1991.

[15]J. A. Meijerink and H. A. Van der Vorst. Guidelines for the usage of incomplete decompositions in solving sets of linear equations as they occur in practical problems. *J. Comp. Phys.*, 44:134–155, 1981.

A CELL-CENTERED MULTIGRID ALGORITHM FOR ALL GRID SIZES*

Thor Gjesdal
Christian Michelsen Research A/S
N-5036 Fantoft, Norway

SUMMARY

Multigrid methods are optimal; that is, their rate of convergence is independent of the number of grid points, because they use a nested sequence of coarse grids to represent different scales of the solution. This nesting does, however, usually lead to certain restrictions of the permissible size of the discretised problem. In cases where the modeler is free to specify the whole problem, such constraints are of little importance because they can be taken into consideration from the outset. We consider the situation in which there are other competing constraints on the resolution. These restrictions may stem from the physical problem (e.g., if the discretised operator contains experimental data measured on a fixed grid) or from the need to avoid limitations set by the hardware. In this paper we discuss a modification to the cell-centered multigrid algorithm, so that it can be used for problems with any resolution. We discuss in particular a coarsening strategy and choice of intergrid transfer operators that can handle grids with both an even or odd number of cells. The method is described and applied to linear equations obtained by discretisation of two- and three-dimensional second-order elliptic PDEs.

INTRODUCTION

Multigrid methods have during the last decades developed into an important tool in many areas of scientific computation. Because they use a nested sequence of grids to represent different scales of the solution, the multigrid algorithms are optimal in the sense that their computational complexity is linearly proportional to the total number of unknowns in the discretised problem. This nesting does however lead to certain restrictions on the permissible size of the discrete problem, similar to those encountered in other efficient 'divide-and-conquer' algorithms such as the fast Fourier transform or cyclic reduction. In a standard multigrid algorithm, coarsening is usually

*This work was supported by the Research Council of Norway through Grant number 100556/410 and program number STP–30074.

performed by doubling the mesh-spacing. The number of cells in the grid will then be of the form $n_l = C_l 2^k$, where n_l is the number of cells in direction l and C_l is some suitable (small) integer. Early applications of multigrid methods for general grid sizes consisted of padding the fine grid with empty cells. Such padding can lead to potential large overheads in storage requirements and computational complexity. Dendy [1] and Adams [2] have both described modifications of vertex-centered multigrid algorithms that are extended to handle general grid-sizes. In cases with an odd number of cells, Dendy employs a dummy point on the coarse grid, while Adams has devised a special coarsening strategy using a uniform grid spacing at all levels.

Often the modeler is free to specify the whole problem, and then such constraints are of no importance because they can be taken into consideration from the outset. We consider here the situation in which there are other competing constraints on the resolution. These restrictions may stem from the physical problem (e.g., if the discretised operator contains experimental data measured on a fixed grid) or from the need to avoid limitations set by the hardware. We believe that these restrictions must be overcome if the multigrid methods are ever to become a standard inventory in the modeler's toolbox.

In this paper we discuss a modification to the cell-centered multigrid algorithm, so that it can be used for problems with any resolution. The cell-centered algorithm is attractive because cell-centered discretisations are in widespread use, and cell-centered multigrid also has the ability to handle problems with discontinuous or rapidly varying diffusion coefficients using standard grid transfer operators [3, 4, 5].

In the next section we will describe the method with special emphasis on the grid coarsening and construction of the intergrid transfer functions. We will apply the method to linear equations obtained by discretisation of two- and three-dimensional second-order elliptic PDEs and show that the convergence rates are indeed independent of the grid size (even grids with an odd number of cells).

MULTIGRID ALGORITHM

Two level algorithm

To describe the method, we will consider a two-level algorithm for the discretised problem

$$Au = b,$$

where A is the discretised differential operator, which we assume is linear. The two-level algorithm consists of a smoothing step and a correction step where the update to the solution is calculated on a coarse grid. The two components of the multigrid algorithm are complementary; that is, smoothing is used to reduce high frequency error components, while the coarse grid correction is good at eliminating low frequencies in the error. We will denote coarse grid quantities by an overbar, and we can then write the algorithm in symbolic form as

$$M = S^{\nu_2}(I - P\overline{A}^{-1}RA)S^{\nu_1},$$

Level	5	4	3	2	1
$nx \times ny$	64×64	32×32	16×16	8×8	4×4
	65×17	33×9	17×5	9×5	5×5

Figure 1: Multigrid hierarchy for five-level system to illustrate grid coarsening strategy.

where M is the two-level error reduction operator; R, P are the restriction and prolongation operators, and S is the smoothing operator with ν_1, ν_2 the number of pre- and post-smoothing sweeps, respectively. We obtain the multigrid algorithm by recursive application of the two-level algorithm to solve the coarse level defect equation $\overline{A}\overline{e} = Rr = R(b - Au)$.

Grid coarsening

For a given fine grid, we choose the coarse grid size as

$$\overline{n_l} = \lfloor n_l/2 \rfloor + \mathrm{mod}(n_l, 2) \tag{1}$$

where $\lfloor \cdot \rfloor$ is the floor function. Standard coarsening, or coarsening in all coordinate directions, is performed as far as possible. For rectangular grids, semi-coarsening is then continued until the coarsest grid has a small number of cells in each direction. In this way, a coarse grid hierarchy is defined for any fine grid, and multigrid iterations can be performed. To illustrate the coarsening strategy, figure 1 shows an example of two five-level systems. The first example shows the standard case with a suitable number of cells and full coarsening in four levels. In the second example we have 'bad' numbers (an odd number of cells in both directions and a moderately rectangular grid). Then we apply full coarsening for two levels and continue semi-coarsening for two levels to obtain a small system on the coarsest grid.

Transfer operators

In this section will we describe the restriction and prolongation operators. For simplicity we will concentrate on the one-dimensional operators. We will then describe briefly how we obtain the higher dimensional operators.

The grid transfer operators must satisfy the well-known accuracy requirement

$$m_R + m_P > 2M,$$

where m_R and m_P are the order of the restriction and the prolongation, respectively, and $2M$ is the order of the differential operator. The order of the grid transfer operators and this rule can be determined either by considering how the interpolation acts

on the Fourier components (Brandt [6], Hemker [7]) or the order of the polynomials used in the interpolation rule (Hackbusch [8]). If we consider second order elliptic operators, we will use a restriction based on linear interpolation, which gives $m_R = 2$, and for the prolongation we will use piecewise constant interpolation ($m_P = 1$). This seems to be more robust than the opposite alternative ($m_R = 1, m_P = 2$) [9].

Prolongation, or coarse-to-fine interpolation, is performed by cell-based piecewise constant interpolation; that is, the coarse grid function values are transferred directly to the fine-grid points that belong to each coarse grid cell.

The fine-to-coarse restriction is defined by the average

$$\overline{u}_i = (Ru)_i = \sum_j R(i,j) u_{\alpha i+j}, \quad \alpha = \lceil n/\overline{n} \rceil . \tag{2}$$

The one-dimensional restriction operator is given by the adjoint of linear interpolation. In the standard case, where n is an even number, this restriction is simply given by the stencil

$$R = \tfrac{1}{4} \left[\begin{array}{cccc} 1 & 3 & 3 & 1 \end{array} \right] .$$

In general, when n is either odd or even, we can envisage two methods to construct the restriction operator. First, we can adopt an approach akin to that of Adams [2] and assume that both the coarse and the fine computational grids are given as a uniform distribution of cells on the unit interval, with spacing $h = 1/n$. The restriction weights can then be calculated by

$$R(i,j) = \max \left\{ 0, 1 - \left| (2i + j - \tfrac{1}{2})h - (i - \tfrac{1}{2})\overline{h})/\overline{h} \right| \right\} . \tag{3}$$

This will give a three- or four-point stencil in all cases. These restriction weights should be scaled so that they add up to the ratio $n/\overline{n}$ [10]. If the fine grid has an even number of cells, this formula will reproduce the standard stencil.

This approach will unfortunately not produce a stable coarse-grid operator when we use the Galerkin approximation, and as a consequence the convergence rate of the method will deteriorate. We have therefore instead developed a restriction operator based on true cell-based coarsening. In this case, one cell at the boundary will be identical to the boundary cell at the finer level, as illustrated in figure 2. A similar aproach was suggested by Hutchinson and Raithby [11] in connection with the use of a low-order restriction operator. For a restriction based on the adjoint of linear interpolation we must modify the stencil in the cells close to the boundary. We get

$$\begin{array}{lcl} R(n-1,:) & = & [\ \tfrac{1}{4} \quad \tfrac{3}{4} \quad 1-w \quad 0 \ ], \\ R(n,:) & = & [\ w \quad 1 \quad 0 \quad 0 \ ], \end{array}$$

where $w = a/b$. If k is the number of immediately preceding finer levels that has an odd number of cells, a and b are given by

$$\begin{array}{ll} a_1 = 1 & a_k = 2a_{k-1}, \\ b_1 = 3 & b_k = 2b_{k-1} - 1. \end{array}$$

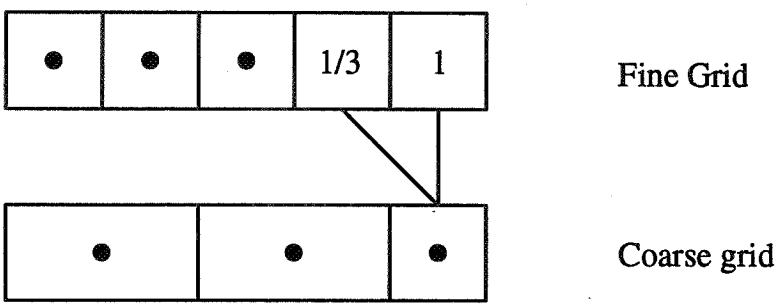


Figure 2: Fine and coarse cells at boundary when the fine grid has an odd number of cells. The numbers indicate the restriction weights for the end point.

When semicoarsening is used, a direction exists in which $\overline{n} = n$. In this case $\alpha = 1$, and both the restriction and the prolongation are given by the identity operator $I = [\ 0 \quad 1 \quad 0\ ]$.

In the multidimensional case, the stencils for restriction and prolongation are determined by tensor products of the one-dimensional stencils. If we let $\mathbf{i}$ and $\mathbf{j}$ denote multi-indices, we will for example have for the restriction stencil in 3D

$$R^{3D}(\mathbf{i},\mathbf{j}) = R^x(i_1,j_1)R^y(i_2,j_2)R^z(i_3,j_3). \tag{4}$$

In other words, in two- and three dimensions restriction will be given by the adjoint of bi- and tri-linear interpolation, respectively.

Coarse grid approximation

There coarse grid matrices are determined by the use of the Galerkin approximation $\overline{A} = RAP$. The Galerkin coarse grid approximation is preferable to straightforward discretisation, because the coarse grid operator can be automatically calculated from the fine grid stencils. This can give the multigrid solver the appearance of a black-box solver where the user only has to supply the coefficients of the discretised equations.

Because the restriction operator is based on bi- and tri-linear interpolation in higher dimensions, the coarse grid stencil will be full (9 points in 2D, 27 points in 3D.) The stencil elements can readily be calculated by the algorithm given by Wesseling [10].

Smoothing

A point that should be noted is that the coarsening strategy we described in the previous section may change the (an-)isotropy of the coarse-grid operator compared

to the operator on the fine grid. This might have to be taken into consideration when we select the smoother. In two dimensions the alternating line Gauss-Seidel method is a robust smoother that is not too expensive. In practice its performance is often quite comparable to Red-Black point relaxation even for isotropic operators. In 3D, the only really robust smoother is alternating plane relaxations, which unfortunately is rather expensive even if a multigrid method is used, to solve the two-dimensional planes. It is therefore difficult to recommend this smoother without reservation. If we have enough knowledge of the problem at hand to decide that a line relaxation method would suffice, a potential gain can be harvested, but for a truly black-box solution plane relaxation is probably the safest bet.

Implementation aspects

In this section we will discuss briefly some practical aspects of the algorithm. The use of one-dimensional interpolation rules makes the implementation fairly modular, and by using features such as derived data types and dynamic memory allocation that are now available in Fortran we have written a combined two- and three-dimensional solver where the PDE can be discretised on any *compact* stencil (the most common are 5, 7, or 9 points in 2D and 7, 12, 19, or 27 points in 3D). With the use of recursion, it is also possible that the solver calls itself for plane smoothing in 3D problems.

Depending on how restriction and prolongation are treated, a modest overhead related to the transfer operations will be realized. In our implementation, this overhead is on the order of $nx \times ny \times nz$ floating point operations (roughly equivalent to one residual calculation on the fine grid) per iteration and $nx+ny+nz$ memory locations. The overhead related to work can however be eliminated if all stencil elements are precomputed and stored. This option will of course lead to a larger storage penalty.

COMPUTATIONAL EXAMPLES

In this section we will demonstrate the convergence of the method for some selected test examples.

Laplace/Poisson equation

First we consider the Laplace equation on rectangular regions with Dirichlet or Neumann boundary conditions.

$$\begin{aligned} \nabla^2 u &= 0, \quad \mathbf{x} \in \Omega \\ u &= 1, \quad \mathbf{x} \in \partial\Omega \quad \text{or} \\ \frac{\partial u}{\partial n} &= 0, \quad \mathbf{x} \in \partial\Omega \end{aligned}$$

Table 1: Two-Dimensional Laplace Equation with Dirichlet Boundary Conditions

Grid size	Levels	Reduction factor	Iterations
31×31	4	0.051	10
32×32	4	0.052	10
33×33	4	0.052	10
50×50	5	0.050	10
63×63	5	0.074	11
64×64	5	0.053	10
65×65	5	0.059	10
99×99	6	0.061	10
150×150	6	0.053	10
65×33	5	0.058	10
101×50	6	0.053	10

2D and 3D calculations with a uniform fine grid

The first set of calculations is performed on the Laplace equation with Dirichlet boundary conditions in a case in which the grid spacing is the same in each direction, so that the fine-grid operator is isotropic. The iterations start off from random initial values in the unknowns and are performed until the residual norm is reduced by a factor of 10^{-12}. The average residual reduction rate, κ, is defined as

$$\kappa = \left(\frac{\|r^n\|_2}{\|r^0\|_2} \right)^{1/n}$$

Table 1 shows results of two-dimensional calculations using alternating line Gauss-Seidel as the smoother for a series of different grid sizes. The results are given for a V(0,1) (sawtooth) cycle. We see from the table that the method works equally well for problem sizes that include both 'good' and 'bad' multigrid numbers.

Results of three-dimensional calculations are given in table 2. The results indicate that the alternating line smoother is sufficiently robust to handle cases in which either odd-numbered grids or semi-coarsening lead to anisotropy in the coarse grid problems

Table 2: Three-Dimensional Laplace Equation with Dirichlet Boundary Conditions

Grid size	Line GS			Plane GS		
	Reduction	Iterations	time	Reduction	Iterations	time
$31 \times 31 \times 31$	0.036	9	0.93	0.016	7	5.36
$32 \times 32 \times 32$	0.031	8	1.00	0.015	7	5.67
$33 \times 33 \times 33$	0.031	8	1.07	0.029	8	7.73
$32 \times 15 \times 19$	0.034	8	0.29	0.014	7	1.81

Table 3: Two-Dimensional Laplace Equation with Homogeneous Neumann Boundary Conditions on a Stretched Grid

Stretching factor:	1.0		1.2		100	
Grid size	κ	n	κ	n	κ	n
8×8	0.139	8	0.136	8	0.024	5
16×16	0.169	9	0.153	8	0.067	7
32×32	0.210	9	0.221	9	Diverge	

as long as the problem on the fine grid is isotropic. The dramatic slow-down seen in the case where we use alternating plane relaxation may be caused by the start-up overhead of the multigrid solver. A way to alleviate this might be to rework the plane-smoother to precompute the coefficients in all the planes. This will however lead to a considerable storage overhead. Another alternative is to investigate whether the three-dimensional coefficients that are already computed can be used also in the plane solver.

Examples with a non-uniform grid

In this section, we will study the effect of anisotropy by introducing a nonuniform grid. Botta and Wubs [12] have shown that solution of partial differential equations on a stretched grid can pose a challenging test case for iterative methods. One of their test cases consists of the two-dimensional Laplace equation on the unit square with homogeneous Neumann boundary conditions. The initial field is given by $f = x^2(1-y)^2$, and the convergence criterion used is that the absolute error should be below 10^{-6}. The grid is generated by geometric stretching so that $s = h_{i+1}/h_i$ is constant. We use a V(0,1) cycle and the alternating line smoother; the results are shown in table 3. We note that the method fails to converge for large values of the stretching factor. The experiments indicate that a critical stretching factor exists depending on the grid size, and that iterations will diverge if the stretching is greater than this limit. In practice will we however only encounter moderate stretching, because an appreciable loss of accuracy occurs even for stretching factors larger than, say, $s \sim 5/4$.

We also performed the same experiment in a 3D case with a 32^3 grid, and we noticed that for moderate stretching rates, $s \leq 2$, we obtained essentially no degradation in the convergence using the alternating line smoother. For $s = 5$, we did, however, notice a significant slow-down as expected.

Stone's problem

This problem was introduced by Stone [13] as a test case for the Strongly Implicit Procedure (SIP), which is a relaxation method based on an incomplete LU decomposition. The problem consists of a heterogeneous diffusion problem on the unit square

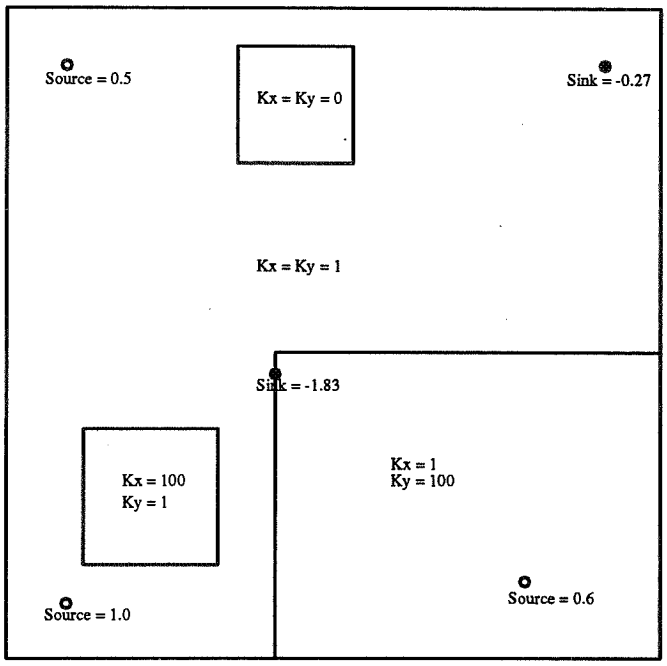


Figure 3: Geometry for Stone's model problem.

given by

$$\begin{aligned} \nabla \cdot \Big[\mathrm{diag}(K_x, K_y) \nabla u \Big] &= f, \quad (x,y) \in [0,1]^2, \\ \vec{n} \cdot \nabla u \Big|_\Gamma &= 0. \end{aligned}$$

The geometry of the problem, specifying the conductivities and the sources, is depicted in figure 3. This problem was solved on a grid with 30×30 cells, using 4 levels in the multigrid iterations. The initial field was identically zero. The convergence factors for this problem are given in table 4.

SIMPLE pressure correction equation

The pressure correction equation in the SIMPLE algorithm for solution of the incompressible Navier-Stokes equations can be interpreted as an elliptic operator in

Table 4: Stone's Model Problem ($\epsilon = 10^{-8}$)

Cycle	Reduction factor	Iterations
V(0,1)	0.24	13
V(1,1)	0.09	8

Table 5: Pressure Correction Equation in the Two Backward Facing Step Examples; Results for the First 20 Outer Iterations

	BFS		BFS-POR	
Iteration	κ	n	κ	n
1	0.100	9	0.125	10
2	0.101	9	0.253	15
3	0.096	8	0.267	15
4	0.090	8	0.234	13
5	0.081	8	0.174	11
10	0.080	7	0.185	11
20	0.096	8	0.179	10
Average	0.089		0.191	

the form

$$D(\gamma G p') = D\mathbf{u},$$

where D and G are discretised divergence and gradient operators, respectively, p' is the pressure correction and $\mathbf{u}$ is the intermediate velocity field. The diffusivity γ consists of the inverse of a diagonalisation of the momentum operator and geometric terms. The equation is usually employed with homogeneous Neumann boundary conditions. The multigrid solver has been used to solve the pressure correction equation for the test case of a backward-facing step at $Re = 800$. The step size was half the channel height, and the channel had a length of 30 step heights. The grid had 30×10 cells, which gives a grid aspect ratio of $5 : 1$.

In the first test, the standard set-up from the benchmark results of Gartling [14] was used. In order to assess the effect of the use of porosities on convergence rates, we performed a second test in which a thin vertical plate with zero porosity and a height equal to the step height was placed in the channel downstream of the two main recirculation bubbles. Table 5 shows the residual reduction rates for the multigrid solver in the two examples. The results show that even though the convergence of the solver deteriorates in the case for which we have zero porosity (and as a consequence, zero diffusivity γ), its performance is still good.

CONCLUSIONS

We have described a generalisation of the cell-centered multigrid algorithm to cover problems with general resolution. Smoothing and the multigrid scheduling are not affected by the extensions, but changes have been made in the grid coarsening strategy and, consequently, in the design of the intergrid transfer operators. We found that cell-based coarsening was better than a mapping of the coarse grid to a uniform mesh. Numerical experiments show that the solver gives multigrid convergence for all

grid sizes in a number of test cases. The alternating line Gauss-Seidel relaxation is a good smoother for the two-dimensional solver. Its performance was also satisfactory in some 3D problems. In some cases involving extreme grid stretching the method seems to fail. This failure is, however, of small practical importance.

REFERENCES

[1] Dendy, J. E., Black box multigrid, *J. Comput. Phys.*, 48:366–386, 1982.

[2] Adams, J. C., MUDPACK 2: multigrid software for approximating elliptic partial differential equations on uniform grids with any resolution, *Appl. Math. Comput.*, 53:235–249, 1993.

[3] Wesseling, P., Cell-centered multigrid for interface problems, *J. Comput. Phys.*, 79:85–91, 1988.

[4] Khalil, M., *Analysis of Linear Multigrid Methods for Elliptic Differential Equations with Discontinuous and Anisotropic Coefficients*, Ph.D. thesis, Delft University of Technology, 1989.

[5] Khalil, M. and Wesseling, P., Vertex-centered and cell-centered multigrid for interface problems, *J. Comput. Phys.*, 98:1–20, 1992.

[6] Brandt, A., Guide to multigrid development, in Hackbusch, W. and Trottenberg, U., editors, *Multigrid Methods*, volume 960 of *Lecture Notes in Mathematics*, pp. 220–312, Springer-Verlag, Berlin, 1982.

[7] Hemker, P. W., On the order of prolongations and restrictions in multigrid procedures, *J. Comput. Appl. Math.*, 32:423–429, 1990.

[8] Hackbusch, W., *Multigrid Methods and Applications*, volume 4 of *Computational Mathematics*, Springer–Verlag, Berlin, 1985.

[9] Ersland, B. G. and Teigland, R., Comparison of two cell–centered multigrid schemes for problems with discontinuous coefficients, *Numer. Meth. for PDE*, 9:265–283, 1993.

[10] Wesseling, P., *An Introduction to Multigrid Methods*, Pure and Applied Mathematics, John Wiley & Sons, Chichester, 1992.

[11] Hutchinson, B. R. and Raithby, G. D., A Multigrid Method Based on the Additive Correction Strategy, *Numer. Heat Transf.*, 9:511–537, 1986.

[12] Botta, E. F. F. and Wubs, F. W., The convergence behaviour of iterative methods on severely stretched grids, *Int. J. Numer. Meth. Engng.*, 36:3333–3350, 1993.

[13] Stone, H. L., Iterative Solution of Implicit Approximations of Multidimensional Partial Differential Equations, *SIAM J. Numer. Anal.*, 5:530–558, 1968.

[14] Gartling, D. K., A Test Problem for Outflow Boundary Conditions—Flow over a Backward-Facing Step, *Int. J. Numer. Methods Fluids*, 11:953–967, 1990.

Numerical study of multigrid methods with various smoothers for the elliptic grid generation equations

W. L. Golik *
Department of Mathematics and Computer Science
University of Missouri at St. Louis
St. Louis, MO 63121
golik@arch.umsl.edu

Abstract

A robust solver for the elliptic grid generation equations is sought via a numerical study. The system of PDEs is discretized with finite differences, and multigrid methods are applied to the resulting nonlinear algebraic equations. Multigrid iterations are compared with respect to the robustness and efficiency. Different smoothers are tried to improve the convergence of iterations. The methods are applied to four 2D grid generation problems over a wide range of grid distortions. The results of the study help to select smoothing schemes and the overall multigrid procedures for elliptic grid generation.

INTRODUCTION

Numerical grid generation arose from the need to compute solutions of partial differential equations defined over physical domains with complicated geometry. By transforming a physical domain to a simpler computational region (e.g., a square or a cube), the complication of the shape of the domain is removed from the problem. Although the transformed PDEs over the simple region are usually more complicated, they are easier to discretize with finite difference or finite volume methods. The domain transformation can be viewed as an introduction of a general curvilinear grid on the original domain. This explains the name: grid generation.

The basic grid generation problem can be formulated in the following way: given a physical domain $\Omega \in R^d$, a computational domain $U \in R^d$, and a nonsingular parametric mapping $\partial\mathbf{x}$ of the domain boundaries

$$\partial\mathbf{x} : \partial U \rightarrow \partial\Omega$$

extend this mapping to a mapping

$$\mathbf{x} : U \rightarrow \Omega$$

from the computational region to the physical domain. Here d denotes the dimension of the space containing Ω, e.g., $d = 2$ describes planar problems. Such a mapping $\mathbf{x}$ is called a boundary-conforming map, and the map generates a boundary-conforming curvilinear grid in domain Ω.

Elliptic grid generation is one of the popular methods for constructing boundary conforming grids. It constructs the grid mapping $\mathbf{x}$ as the solution of a system of elliptic partial differential equations defined on U subject to the boundary condition satisfying $\mathbf{x}(\partial D) = \partial\Omega$. A major advantage of this approach is that the curvilinear grid in Ω is smooth, which results in small truncation errors in finite difference discretizations of transformed differential equations. A major disadvantage is that the grid construction itself involves a numerical solution of a system of quasi-linear elliptic PDEs and

*supported by 1994 University of Missouri at St. Louis Research Award

requires much longer execution times than other types of grid generation (algebraic, parabolic, or hyperbolic).

ELLIPTIC GRID GENERATION AND DISCRETIZATION

A general elliptic grid generation system of equations can be written as

$$\mathcal{T}\mathbf{x} = \mathbf{F}, \tag{1}$$

with Dirichlet boundary conditions $\mathbf{x}(\partial U) = \partial\Omega$, where $\mathcal{T}$ is a second order, quasi-linear, elliptic differential operator and $\mathbf{F}$ is the inhomogeneous part of the system (usually a first order differential operator).

A widely used elliptic system for grid generation (see [2]) is the inhomogeneous Thompson-Thames-Mastin (ITTM) generator given by the following equations (in two dimensions):

$$\left(g_{22}\frac{\partial^2}{\partial\xi^2} - 2g_{12}\frac{\partial^2}{\partial\xi\partial\eta} + g_{11}\frac{\partial^2}{\partial\eta^2}\right)\mathbf{x} = -g_{22}p\mathbf{x}_\xi - g_{11}q\mathbf{x}_\eta, \tag{2}$$

where $g_{11} = \mathbf{x}_\xi \cdot \mathbf{x}_\xi$, $g_{12} = \mathbf{x}_\xi \cdot \mathbf{x}_\eta$, $g_{22} = \mathbf{x}_\eta \cdot \mathbf{x}_\eta$ are the elements of the metric matrix and p, q are user defined functions which allow some measure of local control of grid cells. This system, together with appropriate Dirichlet boundary conditions, defines the mapping $\mathbf{x} : U \to \Omega$ from the computational to the physical domain. For the derivation of the ITTM equations and other examples, see [2, 3].

Equations (2) may be solved numerically via standard central finite differences on a uniform, Cartesian grid in the computational domain U. Due to the presence of mixed derivatives, the nine-point stencil must be used. Grid points are indexed lexicographically by a two-tuple of integers $i = (i_1, i_2)$, $i_1 = 1, 2, ..., M$, $i_2 = 1, 2, ..., N$. Let $\mathbf{X} = (X^1, X^2)$, where $\mathbf{X}_i$ is the approximation of $\mathbf{x}(\xi_{i_1}, \eta_{i_2})$, be blockwise ordered. Orderings used in multigrid smoothers may be different and are specified with the smoothers. In blockwise ordering we have $(\cdots, X^1_i, X^1_{i+1}, \cdots, X^2_i, X^2_{i+1}, \cdots)$. The discretization results in the following nonlinear algebraic system:

$$\left(G_{22}(\mathbf{X})(D_{11} + PD_1) - 2G_{12}(\mathbf{X})D_{12} + G_{11}(\mathbf{X})(D_{22} + QD_2)\right)\mathbf{X} = F, \tag{3}$$

where G_{kl} are diagonal matrices with symmetric finite difference approximations of the metrics g_{kl} evaluated at the nodes indexed by the two-tuples i. The matrices $D_{11}, D_{12}, D_{22}, D_1$, and D_2 represent symmetric finite difference operators approximating the derivatives; P and Q are diagonal matrices corresponding to the user defined functions p and q, and F is a vector containing the values of the mapping x at the boundary nodes.

The nonlinear system (3) can be solved iteratively in one of several ways. The Newton iterations typically converge faster for problems with mild grid distortions but lead to singular Jacobians and divergent iterations for strong distortions. Lagging the metric coefficients G_{kl} produces a more robust but slower solver. At each step a linear system is to be solved of the form

$$\left(G_{22}(\mathbf{X}^n)(D_{11} + PD_1) - 2G_{12}(\mathbf{X}^n)D_{12} + G_{11}(\mathbf{X}^n)(D_{22} + QD_2)\right)\mathbf{X}^{n+1} = F, \tag{4}$$

where $G_{kl}(\mathbf{X}^n)$ denotes diagonal matrices with symmetric finite differences approximations of the metrics g_{kl} at $(\mathbf{X}^n)$.

A simpler nonlinear iteration also lags the P and Q terms yielding the system

$$\left(G_{22}(\mathbf{X}^n)D_{11} - 2G_{12}(\mathbf{X}^n)D_{12} + G_{11}(\mathbf{X}^n)D_{22}\right)\mathbf{X}^{n+1} = F - \left(G_{22}(\mathbf{X}^n)PD_1 + G_{11}(\mathbf{X}^n)QD_2\right)\mathbf{X}^n. \tag{5}$$

Using blockwise ordering both of these systems can be written in a block diagonal form

$$\mathbf{A}\mathbf{X} = \mathbf{b}, \tag{6}$$

where

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_1 & 0 \\ 0 & \mathbf{A}_2 \end{pmatrix}, \qquad \mathbf{X} = \begin{pmatrix} \mathbf{X}^1 \\ \mathbf{X}^2 \end{pmatrix}, \qquad \mathbf{b} = \begin{pmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{pmatrix}. \tag{7}$$

At each step of the iteration the sparse nonsymmetric linear system (6) is solved using a linear multigrid method with a V-cycle. The initial guess $\mathbf{X}^0$ is provided by an algebraic grid generation algorithm [2, 3].

In this paper we study the performance of various multigrid smoothers for equations (4) and 5).

MULTIGRID

In the multigrid solution of linear systems (equations (4) and (5) we have looked at 17 smoothers. Seven of them were point Gauss-Seidel with various ordering: horizontal forward (PHF), vertical forward (PVF),vertical backward (PVB), horizontal symmetric (PHS), vertical symmetric (PVS), alternating forward (PAF), and alternating symmetric (PAS). Nine other smoothers were line Gauss-Seidel variants: horizontal forward (LHF), vertical forward (LVF), vertical backward (LVB), horizontal symmetric (LHS),vertical symmetric (LVS), alternating forward (LAF), alternating backward (LAB), alternating symmetric (LAS), and alternating forward zebra (LAFZ). The last method considered was the point incomplete LU factorization smoother (PILU). Obviously the above smoothers have different complexity counts per one iteration. Clearly, PVF and PVB have the same complexity as the PHF smoother; PHS, PVS, and PAF require twice as many computations, and the PAF smoother is 4 times as costly. The complexity of line smoothers is as follows: LHF, LVF and LVB smoothers require 11/9 of PHF computations, the LHS, LVS, LAF, LAB, and LAFZ smoothers cost 22/9 times more, and the LAS smoother takes about 4.5 times as long.

The linear multigrid algorithm used the V-cycle with single pre-smoothing and single post-smoothing at each level. The coarsest grid was as coarse as possible: it consisted of one internal point (and eight on the boundary). The Galerkin coarse grid approximation was used, and a direct solver applied on the coarsest level. The prolongation operators were bilinear, and the restriction operators were scaled adjoints of the prolongation operators.

NUMERICAL EXAMPLES AND RESULTS

To measure performance of the algorithms, the reduction factors were used. At each step of nonlinear iterations, n multigrid V-cycles were applied. Define $r = \|\mathbf{R}\|_{frob}/N^3$, where $\mathbf{R} = \mathbf{b} - \mathbf{AX}$ is the residual of equation (6), and $\|\cdot\|_{frob}$ denotes the Frobenius norm. Denoting the norm of the residual after the i-th V-cycle by r_i we define the i-th reduction factor ρ_i by

$$\rho_i = r_i/r_{i-1}$$

and the average reduction factor by

$$\bar{\rho} = (r_n/r_0)^{1/n}.$$

The four test problems, the square, the trapezoid, the L-shaped domain (backstep), and the airfoil, were chosen from the "grid gallery" given in [2]. The domains and some 16x16 curvilinear grids generated by the ITTM equations are given in figure 1.

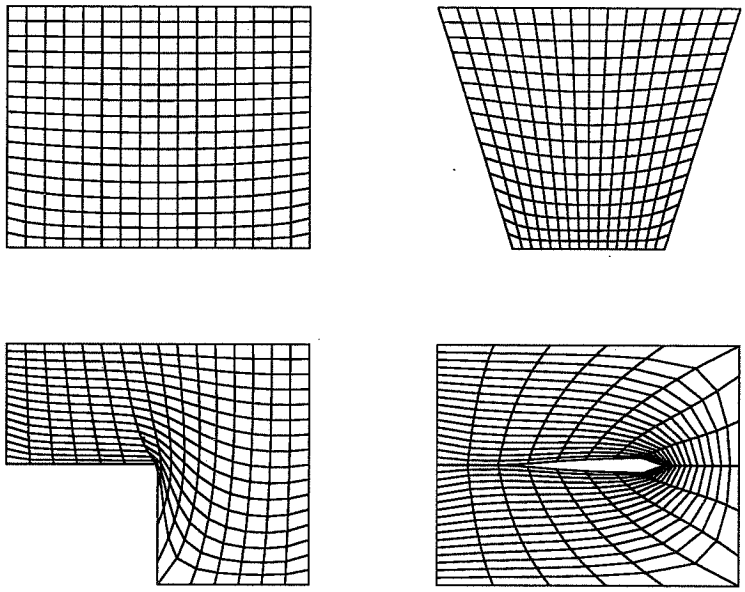


Figure 1. Grids generated on various 16x16 domains: SQUARE, TRAPEZOID, BACKSTEP, and AIRFOIL with elliptic generators.

The first and second problems are very simple with small mixed derivatives. Problems three and four are harder due to high values of mixed derivatives in some parts of the computational domain. The grid control functions p and q in the ITTM equations were chosen so that the grid cells were concentrated at the "bottom" of the domains. In the case of the airfoil the physical domain is doubly connected. A cut emanating from the tail of the airfoil enables it to be mapped onto a computational square. The bottom of the square corresponds to the surface of the airfoil.

We performed a series of numerical experiments for all four test problems by varying the parameters of the control functions. First, the reduction factors of 25 V-cycles with different smoothers were measured for the first nonlinear iteration. To determine the best performance, the relaxation parameter was varied from 0.1 to 1.9 by 0.1. Tables 1-4 give the reduction factors from the first 3 V-cycles and the average reduction factor for all 17 smoothers. The listed results are for grid control functions $p = 0$ and $q(\xi, \eta) = -5\exp(-5\eta)$ in the ITTM equations, but the results are typical for a wide range of control function parameters. The multigrid iteration terminated after 25 V-cycles, or after machine accuracy was reached, whichever occurred first. The last ("asymptotic") reduction factor ρ_{as} for each smoother was also given. If the last 3 reduction factors differ by less than 0.0005, then the value of ρ_{as} was marked with an asterisk.

From these tables, we make the following observations:

• All point and line Gauss-Seidel smoothers work for all the test problems. The "optimal" relaxation parameters ω vary significantly as the problem changes, but all of them turn out to be larger than 1 (overrelaxation). However, the multigrid iterations converged for every tested value of $0.1 < \omega < 2$.

• With the PILU smoother, the relaxation parameter is much less sensitive to the changes of the problem and the grid control functions. In fact, the best values of ω were contained between 0.7 and 0.9. However, the PILU smoother was divergent for the airfoil problem (example 4) for any value of ω.

• The decrease of the reduction factor obtained from applying symmetric or alternating Gauss-Seidel smoothers rather than forward (or backward) ones do not seem to justify the computational costs. The reduction factors for the former are larger than the square of the reduction factors for the latter. However, the issue of choosing the best ordering for the problem remains. The smoothers with similar computational complexities have widely different reduction factors in examples 3 and 4.

• Line Gauss-Seidel smoothers perform better than point Gauss-Seidel smoothers of similar complexity on more difficult problems (examples 3 and 4) and nearly the same on easy problems (examples 1 and 2).

With the above observations the best smoothers were tested in the full nonlinear iteration for examples 3 (backstep) and 4 (airfoil). The iterations described in equations (4), and (5) were implemented with initial guesses supplied by an algebraic grid generator. The residuals ρ_i after each nonlinear iteration were computed using the "exact" values of the metric tensor in the ITTM equation. The "exact" metric tensor was computed by running the iterations until machine convergence was reached prior to the actual tests. Fifty iterations were performed, unless the process was interrupted earlier when the residual reached 10^{-10}. The results are contained in tables 5 and 6. The line Gauss-Seidel smoothers can be seen to give faster results.

CONCLUDING REMARKS

The object of the study, of which the preliminary results are reported here, was to select the most robust smoothers for multigrid in elliptic grid generation. Since the shape of the physical domain in the grid generation and the control grid functions usually induce elliptic grid equations with sharply varying coefficients (and possibly large convective terms), the optimal smoothers may be found from among the line ILU methods. We plan to investigate this possibility next. Also we are working on a study of smoothing for the full approximation storage (FAS) [1, 4] to apply multigrid directly to the nonlinear system (3).

REFERENCES

[1] W. HACKBUSCH, *Multi-grid Methods and Applications*, vol. 4, Springer Series in Comp. Math., Springer–Verlag, Berlin, 1985.

[2] P.M. KNUPP AND S. STEINBERG, *The Fundamentals of Grid Generation*, CRC Press, 1994.

[3] J.F. THOMPSON, Z.U.A. WARSI, AND C.W. MASTIN, *Numerical Grid Generation: Foundations and Applications*, North-Holland, Elsevier, New York, 1985.

[4] P.WESSELING,*An Introduction to Multigrid Methods*, John Wiley & Sons, Chichester, 1992.

Table 1: 128x128 SQUARE Reduction Factors with "Optimal" Relaxation

Smoother	ω	ρ_1	ρ_2	ρ_3	l	ρ_{as}	$\hat{\rho}$
PHF	1.1	0.151	0.152	0.155	11	0.144*	0.151
PVF	1.2	0.192	0.149	0.123	12	0.239*	0.164
PVB	1.1	0.204	0.175	0.192	13	0.202*	0.189
PHS	1.3	0.084	0.051	0.056	8	0.087*	0.071
PVS	1.3	0.084	0.051	0.056	8	0.087*	0.071
PAF	1.4	0.065	0.059	0.059	7	0.060	0.060
PAS	1.4	0.027	0.021	0.022	6	0.026*	0.024
LHF	1.0	0.295	0.153	0.129	11	0.133*	0.106
LVF	1.0	0.225	0.152	0.138	11	0.140*	0.140
LVB	1.0	0.225	0.152	0.138	11	0.140*	0.140
LHS	1.1	0.134	0.061	0.052	8	0.057*	0.060
LVS	1.0	0.065	0.053	0.053	8	0.057*	0.056
LAF	1.3	0.051	0.026	0.026	6	0.030*	0.031
LAB	1.3	0.052	0.027	0.028	7	0.032*	0.032
LAFZ	1.0	0.048	0.040	0.049	7	0.053*	0.049
LAS	1.4	0.016	0.010	0.010	5	0.011	0.011
PILU	0.9	0.041	0.031	0.025	6	0.022	0.027

* "asymptotic"

Table 2: 128x128 TRAPEZOID Reduction Factors with "Optimal" Relaxation

Smoother	ω	ρ_1	ρ_2	ρ_3	l	ρ_{as}	$\hat{\rho}$
PHF	1.2	0.237	0.199	0.175	14	0.286	0.214
PVF	1.1	0.238	0.200	0.180	14	0.244	0.206
PVB	1.2	0.279	0.194	0.193	15	0.250*	0.227
PHS	1.3	0.114	0.091	0.111	11	0.154	0.130
PVS	1.3	0.112	0.089	0.108	10	0.146	0.123
PAF	1.3	0.109	0.078	0.076	9	0.099	0.086
PAS	1.4	0.040	0.037	0.044	7	0.053	0.046
LHF	1.1	0.408	0.197	0.195	14	0.266	0.216
LVF	1.0	0.248	0.185	0.171	13	0.186	0.177
LVB	1.1	0.240	0.201	0.200	13	0.198*	0.195
LHS	1.2	0.207	0.098	0.086	10	0.106	0.101
LVS	1.1	0.089	0.088	0.084	9	0.088	0.086
LAF	1.3	0.067	0.039	0.041	7	0.047	0.046
LAB	1.3	0.075	0.046	0.049	8	0.052*	0.053
LAFZ	1.1	0.108	0.058	0.077	9	0.099	0.089
LAS	1.5	0.050	0.015	0.014	6	0.016	0.019
PILU	0.9	0.059	0.032	0.034	9	0.191	0.093

* "asymptotic"

Table 3: 128x128 BACKSTEP Reduction Factors with "Optimal" Relaxation

Smoother	ω	ρ_1	ρ_2	ρ_3	l	ρ_{as}	$\hat{\rho}$
PHF	1.5	0.356	0.349	0.467	24	0.833	0.715
PVF	1.7	0.551	0.456	0.490	24	0.689	0.645
PVB	1.6	0.749	0.428	0.463	25	0.700	0.628
PHS	1.5	0.179	0.281	0.379	25	0.745	0.630
PVS	1.7	0.311	0.394	0.452	25	0.618	0.549
PAF	1.7	0.276	0.321	0.422	23	0.626	0.543
PAS	1.8	0.243	0.318	0.376	25	0.459	0.418
LHF	1.7	0.550	0.451	0.536	25	0.695	0.659
LVF	1.4	0.303	0.258	0.359	25	0.475	0.416
LVB	1.4	0.313	0.306	0.412	24	0.547	0.500
LHS	1.7	0.290	0.441	0.489	25	0.613	0.553
LVS	1.4	0.226	0.193	0.188	22	0.376*	0.311
LAF	1.6	0.179	0.147	0.199	18	0.282*	0.245
LAB	1.6	0.194	0.202	0.220	22	0.358	0.307
LAFZ	1.4	0.723	0.104	0.201	23	0.527	0.465
LAS	1.7	0.081	0.103	0.119	14	0.236	0.153
PILU	0.7	0.148	0.115	0.115	14	0.167	0.151

* "asymptotic"

Table 4: 128x128 AIRFOIL Reduction Factors with "Optimal" Relaxation

Smoother	ω	ρ_1	ρ_2	ρ_3	l	$\rho_a s$	$\hat{\rho}$
PHF	1.6	0.856	0.464	0.515	20	0.683	0.609
PVF	1.6	0.900	0.404	0.433	20	0.685	0.612
PVB	1.6	0.747	0.509	0.472	25	0.825	0.684
PHS	1.7	0.367	0.531	0.494	20	0.522	0.482
PVS	1.7	0.360	0.498	0.480	20	0.501	0.474
PAF	1.7	0.588	0.396	0.463	20	0.467	0.478
PAS	1.7	0.221	0.252	0.233	20	0.406	0.292
LHF	1.6	0.959	0.412	0.425	20	0.712	0.607
LVF	1.1	0.175	0.184	0.171	15	0.226*	0.189
LVB	1.1	0.175	0.184	0.171	15	0.226	0.189
LHS	1.7	0.365	0.513	0.495	20	0.500	0.489
LVS	1.1	0.079	0.057	0.052	10	0.124*	0.083
LAF	1.1	0.119	0.123	0.116	13	0.184*	0.187
LAB	1.1	0.119	0.124	0.118	13	0.185*	0.186
LAFZ	1.2	0.298	0.080	0.097	12	0.132*	0.123
LAS	1.2	0.041	0.032	0.036	9	0.087*	0.056
PILU	n/a	n/a	n/a	n/a	n/a	n/a	n/a

* "asymptotic"

Table 5: Comparison of Smoothing Performance in Nonlinear Iterations for 128x128 BACKSTEP with Initial Residual $r_0 = 4.30e-02$

Smoother	PVB	PVS	LVF	LAF
Equation (4)				
# iter	50	48	45	46
r_{last}	1.12e-10	5.01e-11	9.10e-11	9.65e-11
$\hat{\rho}$	0.674	0.651	0.642	0.649
Equation (5)				
# iter	50	29	29	26
r_{last}	1.32e-10	5.69e-11	4.23e-11	6.45e-11
$\hat{\rho}$	0.676	0.494	0.489	0.458

Table 6: Comparison of Smoothing Performance in Nonlinear Iterations for 128x128 AIRFOIL with Initial Residual $r_0 = 5.85e-02$

Smoother	PHF	PVS	LVF	LVS
Equation (4)				
# iter	50	50	50	50
r_{last}	6.49e-09	8.56e-09	6.80e-09	6.65e-09
$\hat{\rho}$	0.726	0.730	0.727	0.726
Equation (5)				
# iter	50	31	30	28
r_{last}	5.15e-09	9.50e-11	4.58e-11	5.16e-11
$\hat{\rho}$	0.723	0.521	0.497	0.495

SOME ASPECTS OF MULTIGRID METHODS ON NON-STRUCTURED MESHES

H. Guillard and N. Marco

INRIA, Sophia-Antipolis,2004 Route des Lucioles, BP 93
06902 Sophia-Antipolis Cedex (France)
e-mail: Herve.Guillard@inria.fr and Nathalie.Marco@inria.fr

SUMMARY

To solve a given fine mesh problem, the design of a multigrid method requires the definition of coarse levels, associated coarse grid operators and inter-grid transfer operators. For non-structured simplicial meshes, these definitions can rely on the use of non-nested triangulations. These definitions can also be founded on agglomeration/aggregation techniques in a purely algebraic manner. This paper analyzes these two options, shows the connections of the volume-agglomeration method with algebraic methods and proposes a new definition of prolongation operator suitable for the application of the volume-agglomeration method to elliptic problems.

1 Introduction

Unstructured meshes are now a common tool in large scale scientific computing. With respect to structured grids, the use of this type of data representation offers the advantage of larger flexibility in adapting the mesh to complex geometries and complicated solutions. However, this approach also places a larger demand on the design of discretisation methods and solution algorithms. As a matter of fact, classical solvers using the regularity of the mesh may fail or become less efficient on non-structured meshes. Among the solvers that have appeared in the last two decades, multigrid type algorithms have been among the more successful. These methods were originally formulated for structured grids. To run efficiently on non-structured meshes, the solution algorithms have to be adapted or re-formulated. In structured MG algorithms, the building blocks of the methods are the inter-grid transfer and coarse grid operators. The main difficulty with these methods is, thus, to adequately design these operators. Unstructured multigrid algorithms add the additional difficulty on defining the coarse levels. This paper presents some approaches to solving this difficulty.
We first consider geometrical methods that explicitly define a hierarchy of grids. The

simplest method follows a coarse to fine path and generates fine levels starting from a given coarse level. More sophisticated methods generate all levels independently. These last methods place an excessive burden on the mesh generator algorithms, and we will indicate a possible way to automate them. The third geometrical method we will consider is the volume agglomeration MG technique of Lallemand & Dervieux [1] based on finite volume discretisations.
Another possible way to face the difficulty of the generation of coarse levels is to rely on a purely algebraic method. These methods can be interesting because they avoid the geometrical complexities that make the generation of coarse levels tedious in the geometrical approaches. However, these methods are much more difficult to design and analyze. We show, however, that any algebraic method can be interpreted as a geometrical one and as an example analyze in this setting the volume agglomeration MG method. We first show that this method can be viewed as an equation summing technique and then use a geometrical interpretation to analyze some of its deficiencies. We then propose a possible way to improve this method.

2 Geometrical methods

In this section, we consider methods that explicitely define a hierarchy of grids.

2.1 Nested mesh approach

Let Ω be a bounded polygonal domain of the plane, and consider a coarse triangulation $\mathcal{T}_1$ of this domain defined by the set of nodes $\mathcal{N}_1$. For simplicity, we assume that this initial triangulation is regular and quasi-uniform with mesh parameter h_1. Associated with this triangulation, we consider the finite element space of piecewise linear functions $\mathcal{M}_1$. The spaces $\mathcal{M}_j$ will be recursively defined by adding nodes at the midpoints of the edges of the triangles of $\mathcal{T}_{j-1}$ and decomposing each triangle into four congruent triangles. We observe that the regularity and quasi–uniformity constants of the mesh are maintained by this process. Each element of $\mathcal{M}_j$ belongs to $\mathcal{M}_{j+1}$ and thus we obtain a sequence of nested spaces

$$\mathcal{M}_1 \subset \mathcal{M}_2 \subset \cdots \subset \mathcal{M}_n$$

such that $dim\ \mathcal{M}_j = 4\ dim\ \mathcal{M}_{j-1}$. In addition, denoting $h_j = \max_{\tau \in \mathcal{T}_j}\{h_\tau\}$, we exactly have $h_{j+1} = h_j/2$.

To connect the different levels, we need linear operators between them. For this, we first equip each space $\mathcal{M}_j$ by an inner product $(.,.)_j$ defined, for instance by :

$$(u,v)_j = h_j^2 \sum_{x \in \mathcal{N}_j} u(x)\ v(x) \quad \forall (u,v) \in \mathcal{M}_j^2$$

By the quasi–uniformity assumption, this inner product induces a norm denoted by $\|.\|_j$ equivalent to the L^2 norm on $\mathcal{M}_j$. The prolongation operator is simply the identity, and the restriction operator $\mathcal{I}_{j+1}^j$ can be defined by

$$(\mathcal{I}_{j+1}^j\, u_{j+1},\ v_j)_j = (u_{j+1},\ v_j)_{j+1} \tag{1}$$

$\mathcal{I}_{j+1}^j$ is then some kind of L^2 projection on $\mathcal{M}_j$.

We now describe the algebraic counterpart of these definitions.
For each space $\mathcal{M}_j$, we consider the usual nodal basis $\{\varphi^j\}$ defined to be 1 on node $x_i \in \mathcal{N}_j$ and 0 on all other nodes $\in \mathcal{N}_j$. The choice of this basis induces a natural one-to-one mapping between $\mathcal{M}_j$ and $\mathbb{R}^{n_j}(\ n_j = \ dim\ \mathcal{M}_j)$ which we denote as Γ_j:

$$\Gamma_j\ :\ u \in \mathbb{R}^{n_j} \rightarrow \sum_{i=1}^{n_j} u_i\ \varphi_i \in \mathcal{M}_j$$

For simplicity, each space $\mathbb{R}^{n_j}$ is equipped with the scalar product

$$< u, v >_j = h_j^2 \sum_{i=1}^{n_j} u_i\ v_i$$

such that we have

$$< u, v >_j = (\Gamma_j u,\ \Gamma_j v)_j$$

The identity of $\mathcal{M}_j$ considered as an operator from $\mathbb{R}^{n_j}$ into $\mathbb{R}^{n_{j+1}}$ will be denoted by I_j^{j+1} defined by

$$\Gamma_j = \Gamma_{j+1}\ I_j^{j+1}$$

in such a way that the following diagram commutes (see Figure 1.). From definition

$$\begin{array}{ccccc} & & \Gamma_{j+1} & & \\ & \mathcal{M}_{j+1} & \longleftarrow & \mathbb{R}^{n_{j+1}} & \\ Id & \uparrow & & \uparrow & I_j^{j+1} \\ & \mathcal{M}_j & \longleftarrow & \mathbb{R}^{n_j} & \\ & & \Gamma_j & & \end{array}$$

Figure 1: Commutative diagram defining the algebraic prolongation.

(1), we see that the algebraic expression of the restriction operator I_j^{j-1} is given by the scaled matricial transpose of I_{j-1}^j :

$$I_j^{j-1} = \left(\frac{h_j}{h_{j-1}}\right)^2 \left(I_{j-1}^j\right)^t$$

Remark 1: For future reference, we note that in the definition of these operators, the functional spaces are first defined, and then the algebraic expression of the transfer operators are *deduced* from them.

The previous method is very simple and exactly fits into the classical variational multigrid theory. However, it implies a large dependence of the fine mesh node distribution on the coarsest level. Indeed, the mesh division algorithm is a fine mesh generation algorithm, and unfortunately this is a very poor one. Thus, the meshes generated this way are of poor quality (hence, the fine mesh solution will also be of poor quality). Moreover, in many cases, the fine mesh is given, and thus the solvers must be able to deal with an arbitrary given fine mesh instead of building it.

A solution can be to relax the constraint of nestedness of the meshes; this is the non-nested approach that we describe now.

2.2 Non-nested approach

In this approach coarse and fine triangulation are generated independently using any given mesh generators. The solution, residuals, and corrections are transferred back and forth through the different levels using linear interpolation between two successive levels. Thus, now I_{j-1}^{j} represents the linear interpolation between the non-nested spaces $\mathcal{M}_{j-1}$ and $\mathcal{M}_j$, and I_j^{j-1} is its adjoint with respect to the inner products of $\mathcal{M}_j$ and $\mathcal{M}_{j-1}$.

From a practical standpoint, the algorithms are the same regardless of whether or not the triangulations are nested; algebraic operators I_{j-1}^{j} and I_j^{j-1} are needed to transform the internal representation of coarse grid functions expressed in terms of basis functions of $\mathcal{M}_{j-1}$ into their internal representations as fine level functions (in a different basis even in the case of nested spaces). Therefore, in the implementation, little difference exists between the nested and non-nested cases. However, the additional complexity of the non-nested approach appears in the fact that the transfer operators between the different levels are difficult to compute. Regardless of the order of the prolongation, one must determine in what triangles the fine node are located. Thus, this approach requires the use of efficient search algorithms.

We also remark that there are now different choices for the definition of the coarse grid operator. The most natural one is to define it by a re-discretisation of the continuous problem on the coarse grid. This choice preserves the bandwidth of the original operators; however the alternate "Galerkin" definition

$$A^{j-1} = I_j^{j-1} A^j I_{j-1}^{j}$$

can be more efficient (with this definition, the error after an exact correction step is purely in $A_j^{-1}\, Ker\, (I_j^{j-1})$). Of course, the "Galerkin" definition does not preserve the bandwidth of the original operator. In CFD, the non-nested multigrid method appears to be one of the most successful strategies (see, for instance, [2]). However, the need

to generate multiple meshes of the same geometries (that have in addition to respect some ratio between the discretisation parameters of the different levels) results in an excessive burden on the user; a method that relies on the use of many independently generated meshes is simply not practical (especially in 3-D). Consequently, algorithms that consider the generation of the coarse level spaces as part as the solution procedure have to be developed to make these techniques practical. The following method [3] is designed for this task. A recent work by Chan and Smith [4] uses a similar idea.

2.3 Node-nested algorithm

For structured grids, coarse levels are generated by removing one point over two in each coordinate direction and reconnecting the set of remaining nodes. This set of points thus forms a maximal independent subset of the vertices of the fine grid. We recall that a subset S of the vertices of a graph is said to be independent if no two vertices of S are connected by an edge of the graph. An independent subset S is called maximal if adding any additional vertices makes it dependent. It is easily realized that in order to extract a coarse mesh with mesh parameter $\sim 2h$ from a given non-structured one with mesh parameter h, we precisely need to define a maximal independent subset from the vertices of the triangulation. Considering a vertex P in a maximal independent subset S, we see that its nearest neighbours in S are at a distance ~ 2 in the graph. Hence, their *physical* distance from P will be of order $2h$. In practice, it is not always desirable to form a maximal independent set. The major reason is that defining a maximal independent set from the boundary nodes can destroy the actual geometry. Typically in a maximal independent subset, half the boundary nodes are removed. However, some of the boundary nodes are crucial for the description of the geometry and have to be kept at every stage of the coarsening process.

Therefore, the algorithm must first identify these nodes. Considering that the nodes that actually define the geometry are the nodes where the curvature is non-zero, this is done by computing the curvature of the boundary nodes and enforcing that the nodes having curvature above a given threshold are kept on all level during the coarsening process. Once this identification process is performed, the following algorithm can be used:
First, place all the nodes of the current triangulation in a list. Sort this list in such a way that the special nodes defining the geometry are listed first, the boundary nodes are listed next, and the interior nodes are listed last. Then apply :

```
for every node in the list  do

   if node is selected

   then remove all its neighbours

end do
```

This algorithm reduces the number of nodes roughly by a factor of 4 and produces a set of unconnected nodes. A coarse mesh can be obtained from these points by triangulating them. It is in principle possible to use any mesh generation strategy to perform this task; here, a Delaunay-Voronoi method is used.
Figure 2 presents an example of the application of this technique to a triangular mesh around a spark plug.

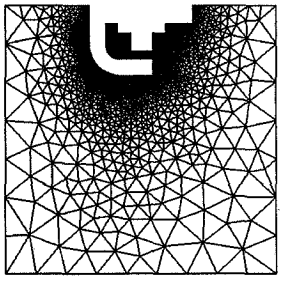


Figure 2.a: Initial level.

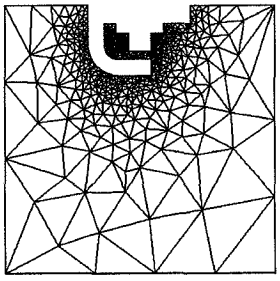


Figure 2.b: After one coarsening.

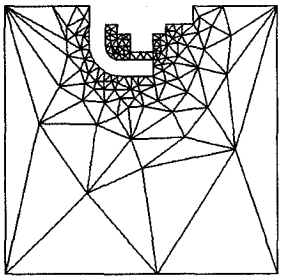


Figure 2.c: Second coarsening.

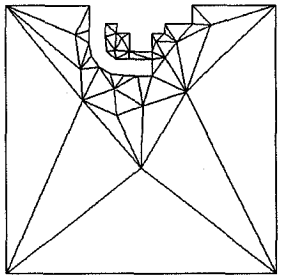


Figure 2.d: Coarser mesh.

Figure 2: Node-nested meshes around a spark plug.

The main advantage of this variation of the non-nested algorithm is that the generation of the coarse levels is part of the solution algorithm; the only input provided

by the user is the fine mesh. In addition, the restriction and prolongation operator can be computed very efficiently, any node of the fine level being itself a coarse node or having at least one neighbor that is a coarse node. For application of this method in CFD see [5] for Euler computations and [5], [6] for Navier-Stokes ones.

2.4 Volume Agglomeration Multigrid method

The previous methods appeal at one stage or another to complex geometrical informations. The Volume Agglomeration MG of Lallemand and Dervieux [1] is an attempt to use only the minimal information given by the connectivity relation of the mesh. This method was originally introduced for first order hyperbolic problems as

$$\frac{\partial}{\partial t} q + \nabla . F(q) = 0 \tag{2}$$

and for finite volume discretisation. Consider, for instance (see Figure 4.a below), the dual control volume mesh of the triangulation of Figure 2.a. To form the coarse grid control volume meshes, neighborings cells are agglomerated together to form a larger coarse cell. This strategy is the exact counterpart of the cell-centered multigrid strategy devised by Wesseling for cell-centered structured algorithms [7]. Figure 3 illustrates this strategy for structured meshes. Figure 4 illustrates it for the

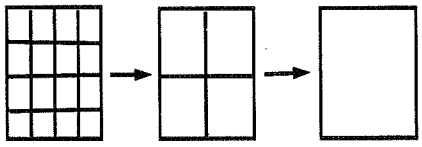


Figure 3: Cell centered multigrid methods.

non-structured mesh of Figure 2. The coarse grids are thus composed of a tiling of the space made of arbitrary polygons. Devising a discretisation on these generalized meshes is not difficult for first-order equations. Application of a finite volume approach results in the same discrete formula as on a regular dual cell mesh:

$$\frac{\partial}{\partial t} \int_{C_i} q + \sum_{j \in \kappa(i)} F(q).\vec{n} dl = 0 \tag{3}$$

and the same numerical flux that is used on the fine mesh can be used to evaluate the integrals. To set up a multigrid strategy, we also need to define the restriction and

prolongation operators. According to the finite volume philosophy where the value associated with a cell can be interpreted as the mean value of the function on this cell, this is done by (we use here the simpler notation $j-1 \rightarrow H$ and $j \rightarrow h$ when only two levels are involved)

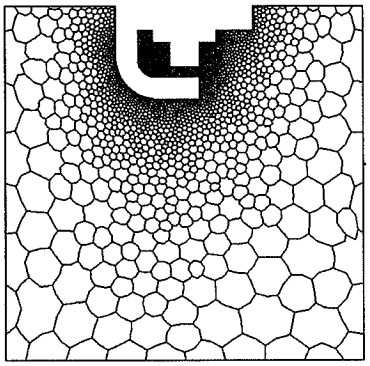


Figure 4.a: Initial level.

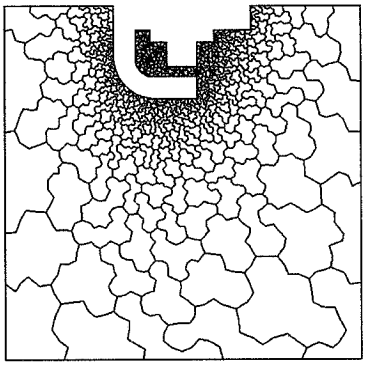


Figure 4.b: After one coarsening.

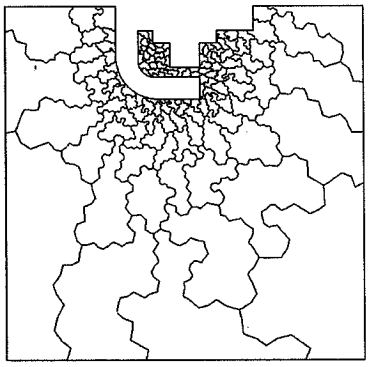


Figure 4.c: Second coarsening.

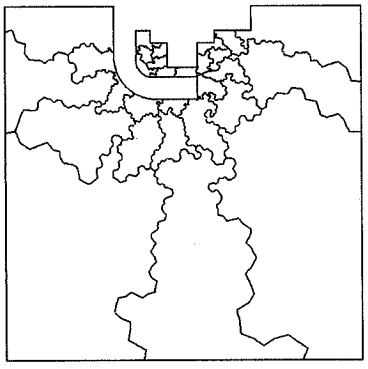


Figure 4.d: Coarser generalized mesh.

Figure 4: Volume agglomerated meshes around a spark plug.

Prolongation

$$I_H^h u_H|_j = u_H|_{l(j)} \tag{4}$$

where $l(j)$ is the coarse cell containing the fine cell whose index is j.

Restriction [1]

$$(I_h^H u_h)_j = \sum_{i \in \mathcal{H}_j} (u_h)_i \tag{5}$$

where $\mathcal{H}_j$ is the set of fine cells that constitute the coarse cell C_j^H.

All the ingredients necessary to set up a multigrid strategy are thus at hand. Application of this technique to the 2-D Euler equations has been done in [1] and in 3-D in [8]. Additional computations have also been done in [9]. All these experiments reveal that the volume agglomeration method is extremely efficient for hyperbolic problems. The generation of the coarse grid is purely automatic and does not require complex mesh manipulations. The computational efficiency of the method is comparable to those of the non-nested approach and of regular structured MG techniques; it is seen that this method largely supersedes the non-nested multigrid methods. However, when applied to an elliptic problem, this technique experiences difficulties. A good way to understand these difficulties is to interpret this method as an algebraic one.

3 Algebraic methods

In the recent past, several attempts have been reported to use only algebraic informations from the discrete problems to be solved. These methods are known as aggregation/disaggregation methods [10] or algebraic multigrid methods [11]. Suppose that we want to solve a linear system on $\mathbb{R}^n$:

$$A\, x = f.$$

where A is a symmetric, positive definite matrix. Let $\{e_i\}_{\{i=1,\ldots,n\}}$ be the canonical basis of $\mathbb{R}^n$, define $\{\mathcal{H}_j\}_{\{j=1,\ldots,P\}}$ to be a partition of $\{1,\ldots,n\}$ into P disjoint sets, and define two vectors $t, z \in \mathbb{R}^n$. Two linear operators between $\mathbb{R}^n$ and $\mathbb{R}^P$ can be constructed by:

$I_h^H : \mathbb{R}^n \to \mathbb{R}^P$: restriction or aggregation operator:

$$I_h^H : \{(U_h)_j\}_{\{j=1,\ldots,n\}} \to \{(V_H)_j\}_{\{j=1,\ldots,P\}} = \{\sum_{k \in \mathcal{H}_j} z_k (U_h)_k\}_{\{j=1,\ldots,P\}} \tag{6}$$

---

[1] It would have been more consistent to define the restriction as $(I_h^H u_h)_j = \dfrac{\sum_{i \in H_j} mes(C_i)(u_h)_i}{\sum_{i \in H_j} mes(C_i)}$. We use here this definition because the interpretation of the volume agglomeration method as an algebraic one yields a simpler expression.

$I_H^h : \mathbb{R}^P \to \mathbb{R}^n$: prolongation or dissaggregation operator:

$$I_H^h : \{(U_H)_j\}_{\{j=1,\ldots,P\}} \to V_h = \sum_{k=1}^{P} (U_H)_k H_k t \tag{7}$$

where H_k is the euclidian orthogonal projection operator onto $\{e_i\}_{\{i \in \mathcal{H}_k\}}$. With the help of these two operators, given a linear operator $L_h \in \mathcal{L}(\mathbb{R}^n, \mathbb{R}^n)$, a coarse grid correction operator belonging to $\mathcal{L}(\mathbb{R}^P, \mathbb{R}^P)$ can be defined by

$$L_H = I_h^H L_h I_H^h \tag{8}$$

It is easy to check that the coefficients of the coarse grid matrix are given by

$$(L_H)_{i,j} = << z, H_i L_h H_j t >> \tag{9}$$

where $<< .,. >>$ is the inner product in $\mathbb{R}^n$. This type of method provides a very general setting to construct multi-level techniques. They are known as aggregation/dissagregation methods and have been introduced for problems in economics or social sciences where they appear in a very natural manner. AMG methods represent an improved variation of these methods, where the way to define the partitions $\{\mathcal{H}_j\}_{\{j=1,\ldots,P\}}$ and the transfer operators are deduced from an analysis of the matrix A itself.

If we now interpret the Volume Agglomeration MG in an algebraic setting and construct the coarse grid operator in the "variational" way, it is easy to see that this technique is equivalent to an equation summing technique. Let L_h be the matrix resulting from the fine grid discretisation, and suppose that L_h is reordered in such a way that

$$L_h = \begin{pmatrix} L_{1,1}, & \ldots, & L_{1,P} \\ \vdots, & \ddots, & \vdots \\ L_{P,1}, & \ldots, & L_{P,P} \end{pmatrix}$$

where $L_{i,j}$ is a $Card(\mathcal{H}_i) \times Card(\mathcal{H}_j)$ block matrix whose coefficients are $l_{p,q}$ for $p \in \mathcal{H}_i$ and $q \in \mathcal{H}_j$. The Volume Agglomeration method results in the choice $t = z = (1, 1, \ldots, 1, 1)^t$, and from (8) we see that the coefficients of the coarse grid matrix L_H are defined by

$$(L_H)_{i,j} = \sum_{p \in \mathcal{H}_i, q \in \mathcal{H}_j} l_{p,q}$$

which exactly corresponds to summing all the entries belonging to the same block. Moreover, if the fine grid matrix results from a nearest neighbor stencil, it will also be the case for the coarse grid one. It is also well known that the "variational" way to construct coarse grid operators implies the preservation of the M-matrix property of the fine grid one.

The analysis of this type of algebraic technique is extremely difficult because no reference is made to the differential equation to be solved or to the mesh on which the

solution is sought. We note, however, that if the original problem has a differential background it is easy to recover functional information from any definition of algebraic transfer operators. For simplicity we consider that the restriction operator is defined as the adjoint of the prolongation operator and that the inner product on $\mathbb{R}^P$ is inherited from the one in $\mathbb{R}^n$ in the following sense:

$$< U_H, V_H >_H = < I_H^h U_H, I_H^h V_H >_h$$

This will allow the algebraic theory to fit into the variational framework. We now simply invert the direction of the diagram displayed in Figure 2 to realize that any algebraic definition of a prolongation operator is equivalent to an *implicit* definition of a coarse grid space $\mathcal{M}_H$ by setting

$$\Gamma_H = \Gamma_h I_H^h \ : \ \mathbb{R}^P \rightarrow \mathcal{M}_h$$

and

$$\mathcal{M}_H = \mathcal{R}(\Gamma_H)$$

Thus one has naturally $\mathcal{M}_H \subset \mathcal{M}_h$ with continuous injection, and we recover the

$$\begin{array}{ccccc} & & \Gamma_h I_H^h(\mathbb{R}^P) & \subset & \mathcal{M}_h \\ \Gamma_H & & \uparrow & & \uparrow \quad \Gamma_h \\ & & \mathbb{R}^P & \underset{I_H^h}{\longrightarrow} & \mathbb{R}^n \end{array}$$

Figure 5: Diagram defining the coarse spaces.

framework of the nested spaces variational theory.

Remark 2: Here, we note that we have first defined the transfer operator and then have *deduced* the definition of the functional spaces associated with them. This is exactly the converse of remark 1.

Moreover it is easy to get an explicit form of a basis of $\mathcal{M}_H$ by:

Proposition: Let $\{e_i\}_{i=1\cdots P}$ be the canonical basis of $\mathbb{R}^P$, then the family $\{\Gamma_H(e_i)\}_{i=1\cdots P}$ is a basis of $\mathcal{M}_H$.

The previous remark can help in the design or analysis of multigrid algorithms. Using it to analyse the volume agglomeration method applied to elliptic equations, we

consider the fine level discretisation space as defined by the usual nodal basis function ϕ_k and we get:
Proposition: The Volume Agglomeration MG method is equivalent to a nested variational method where the coarse grid space is generated by the basis functions:

$$\phi_j^H = \sum_{k \in \mathcal{H}_j} \phi_k \quad \forall j = 1, \ldots, P \tag{10}$$

We see that the coarse grid space $\mathcal{M}_H$ is a very poor one; it does not even contain linear functions. This space is, thus, not dense in H^1, and this implies that the solution of the coarse grid problem is a very poor approximation of the fine grid solution. For instance, consider the 1-D case and define L_h as the usual three point finite difference approximation of the Laplacian, *i.e.* in stencil notation $L_h = \frac{1}{h^2}[-1, 2, -1]$. With the notations of Figure 6, the prolongation operator I_H^h is defined by

$$(I_H^h(U_H))_{2i} = (U_H)_i \quad \text{and} \quad (I_H^h(U_H))_{2i+1} = (U_H)_i \tag{11}$$

with (9), the coarse grid problem writes

$$\frac{1}{H^2}[-1, 2, -1]U_H = \frac{1}{2}\frac{(R_h)_{2i} + (R_h)_{2i+1}}{2} \tag{12}$$

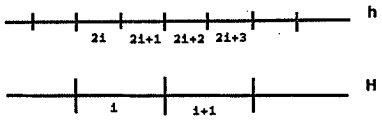


Figure 6: VA-MG in 1-D.

As already noted in [12], a factor 2 is missing in the right-hand side, and the coarse grid operator is not a consistent approximation of the Laplacian. These problems are also well known in structured cell centered multigrid methods (see [7]). In the recent past, several attempts have been proposed to overcome this problem. In [12], the analysis of the 1-D example given previously was extended to 2-D structured meshes, and it was shown that a simple scaling allows a consistent approximation of the Laplace operator to be recovered. The coarse grid problems were then scaled by a factor of 2^k where k is the level number. This strategy gave good results. It has been recently used in [13] for 2-D steady viscous flows with $k - \epsilon$ turbulence modelling and found to be rather efficient from a practical point of view. An alternate approach has also been proposed in [9], where the prolongation operator is choosen by an Algebraic Multigrid heuristic and the coarse grid operator is defined by the variational method. However in practice, this strategy was too costly, and the numerical results reported in [9] use the same scaling strategy as in [12].

4 A prolongation operator for Volume Agglomeration MG

For structured cell centered multigrid methods a simple remedy for the above problem consists in defining prolongation and restriction operators that have higher orders of accuracy. For instance in the 1-D case, the definition (11) is replaced by

$$\begin{array}{l}(I_H^h(U_H))_{2i} = \frac{3}{4}(U_H)_i + \frac{1}{4}(U_H)_{i-1} \\ (I_H^h(U_H))_{2i+1} = \frac{3}{4}(U_H)_i + \frac{1}{4}(U_H)_{i+1}\end{array} \tag{13}$$

This exactly corresponds to a linear interpolation between the barycenters of the coarse cells.

For finite element nonstructured meshes, a similar approach can be considered. The set of the coarse cells $C_i^H (i = 1, \ldots, n)$ constitute a tiling of the domain Ω. If we associate with each coarse cell C_i^H a unique point $i \in C_i^H$ (for instance, the gravity center of the cell), we can triangulate this set of points by any convenient mesh generator algorithm. Although this approach will certainly be efficient, we see that it has few advantages against the node-nested method. With the objective of keeping the amount of geometrical information as small as possible, we try here a simplified variation of this approach that seems to give good results.

Thus, let J_H^h be a prolongation operator having the necessary degree of accuracy. For instance take J_H^h as being the operator defined by a triangulation of the gravity centers of the coarse cells. Then, there exists an $n \times n$ operator a_h such that $J_H^h = a_h I_H^h$, where I_H^h is the straight injection (4). In a finite volume framework, I_H^h is the operator that takes a constant by cell function on the coarse grid and returns the same constant by cell function on the fine grid. On the other hand, J_H^h takes the same constant by cell function on the coarse grid but returns a piecewise linear function. Thus, a_h can be interpreted as a *reconstruction* operator; it transforms a constant by cell function in a piecewise linear function. As an example, let us consider the 1-D case: J_H^h is defined by expression (13), while I_H^h is given by (11). (See figure 6.) On the interval defined by the gravity center of the coarse cells i and $i+1$ the operator a_h transforms the piecewise constant function whose values are U_H^i on cell i and U_H^{i+1} on cell $i+1$ into the linear function

$$u(x) = U_H^i + (x - x_{i+1/2})(U_H^{i+1} - U_H^i)/2h$$

and a_h represents the interpolation of this function on the fine grid. From (11) and (13), it is readily seen that the expression of a_h is

$$\begin{array}{l}(a_h u)_{2i} = \frac{3}{4}u_{2i} + \frac{1}{4}u_{2i-1} \\ (a_h u)_{2i+1} = \frac{3}{4}u_{2i+1} + \frac{1}{4}u_{2i+2}\end{array}$$

or

$$(a_h u)_k = \frac{1}{2}u_k + \frac{1}{4}u_{k+1} + \frac{1}{4}u_{k-1}$$

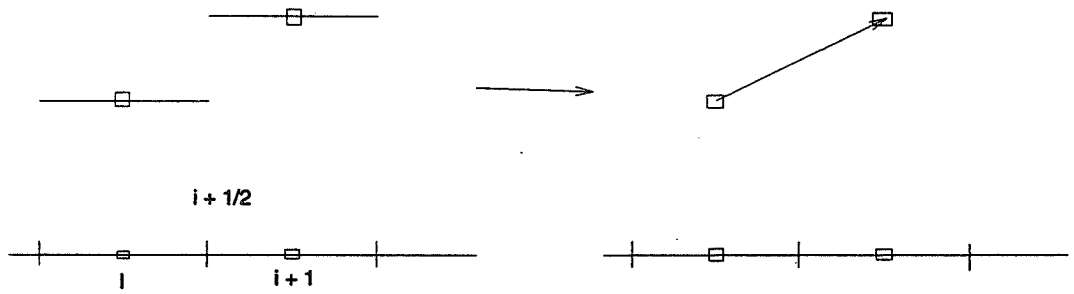


Figure 7: 1-D reconstruction procedure.

Two-dimensional structured cases also reveal that a_h can be interpreted as a reconstruction operator and can be written as

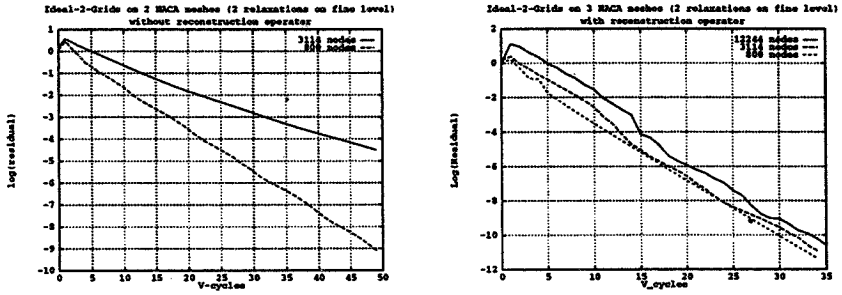
$$(a_h u)_k = \sum_{i \cup \kappa(k)} \alpha_i u_i$$

where $\kappa(k)$ is the set of neighbors of k, and the α_i are coefficients that depend on the geometry.
We then propose to use the same strategy for non-structured meshes. In order to obtain a formula as easily as possible, we define the operator a_h by

$$a_h u_k = \frac{\sum_{i \cup \kappa(k)} vol(C_i) u_i}{\sum_{i \cup \kappa(k)} vol(C_i)} \tag{14}$$

That is, we replace the geometric coefficients α_i by a very crude weighting. Although rather heuristic, this approach seems to give good results. We now report numerical results for solving the Laplace equation on a non-structured triangular mesh around a NACA airfoil with homogeneous Dirichlet boundary conditions.

Figure 8.a shows the convergence curves on three different meshes (i.e. 800, 3000 and 12000 nodes) obtained by a two grid method with full solution of the coarse grid problem and two Jacobi relaxations on the fine mesh. The straight injection (11) is used in this experiment. It is clear that the convergence factor becomes worse as the size of the mesh increases. On the other hand, Figure 8.b shows the results obtained for the same experiment using the improved prolongation (14). The convergence factor is much better, and it is clear that mesh-independent results are obtained. Finally, the same experiment is performed with improved prolongation (14) in a V-cycle setting using 7 different levels for the 12000-node triangulation, 6 for the 3000-node triangulation and 5 for the 800-node triangulations (Figure 9). Again it is seen that mesh independent results are obtained and that there is no decrease in the performance with respect to the 2-grid case.



a: Without the improved prolongation. b: With the improved prolongation.

Figure 8: Two-grid cycle on three different meshes (800, 3000 and 12000 nodes) without and with the improved prolongation.

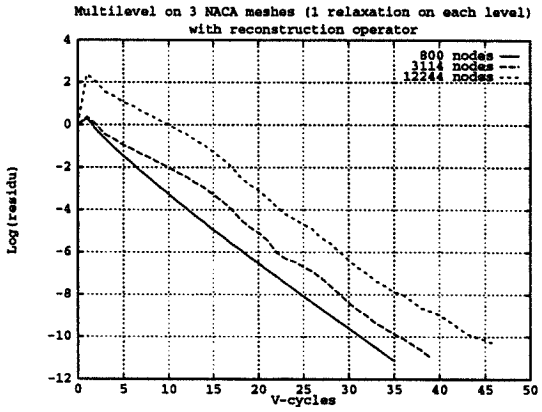


Figure 9: V-cycle with the improved prolongation.

5 REFERENCES

[1] M-H. Lallemand, A. Dervieux, "A multigrid finite element method for solving the two-dimensional Euler equations," Proceedings of the third Copper Mountain conference on Multigrid methods, Lecture Notes in Pure and Applied mathematics, S. F. McCormick, ed. Marcel Dekker, Inc., April 87, pp 337-363.

[2] D. Mavriplis, A. Jameson, L. Martinelli, Multi-grid solution of the Navier-Stokes equations on triangular meshes, ICASE Rpt No 89-11, 1989.

[3] H. Guillard, "Node-nested multigrid method with Delaunay Coarsening," INRIA Rpt No 1898, May 1993.

[4] T. Chan, B. Smith, "Domain decomposition and multigrid algorithms for elliptic problems unstructured meshes, " Electronic Transactions in Numerical Analysis, **2**, pp 171-182, 1994.

[5] E. Morano, A. Dervieux, H. Guillard, M-P. Leclercq, B. Stoufflet, "Faster Relaxations for non-structured MG with Voronoi coarsening," in Proceedings of CFD'92,**vol I**, pp 69-74, (Hirsch et al. eds.), Elsevier 1992.

[6] E. Morano, A. Dervieux "Steady Relaxations Methods for Unstructured MG Euler and Navier-Stokes solutions" Comp. Fluid. Dyn., **5**, No 3-4, pp 137-167, 1995.

[7] P. Wesseling, An introduction to Multigrid methods, Wiley, 1992.

[8] M-H. Lallemand, H. Steve, A. Dervieux, "Unstructured multigridding by volume agglomeration: current status,"Computers and Fluids, **21**, pp397-433, 1992.

[9] V. Venkatakrishnan, D. Mavriplis, "Unstructured multigrid for the 3-D Euler equations," AIAA paper 94-0069, January 1994.

[10] F. Chatelin, W. L. Mirankar, "Acceleration by Aggregation of successive Approximation methods," Linear Algebra and its Applications, **43**, pp 17-47, 1982.

[11] J. Ruge and K. Stuben, Algebraic Multigrid Methods in Multigrid Methods (S. F. McCormick ed), Frontiers in Applied Math, **Vol 3**, SIAM, Philadelphia, pp 73-130, 1987.

[12] B. Koobus, M-H. Lallemand, A. Dervieux, "Unstructured Volume Agglomeration MG, Solution of the Poisson equation," Int. J. for Numerical Meth. in Fluid, **18**, pp 27-42, 1994.

[13] G. Carré, A. Dervieux, "Unstructured multigrid for $k-\epsilon$ kernel," Workshop on Efficient Models for Aeronautics, Manchester, November 11-14, 1994.

SCHWARZ METHODS: TO SYMMETRIZE OR NOT TO SYMMETRIZE[1]

Michael Holst
Applied Mathematics 217-50
Caltech, Pasadena, CA 91125

Stefan Vandewalle
Applied Mathematics 217-50
Caltech, Pasadena, CA 91125

SUMMARY

A preconditioning theory for Schwarz methods is presented. The theory establishes sufficient conditions for multiplicative and additive Schwarz algorithms to yield self-adjoint positive definite preconditioners. It allows for the analysis and use of non-variational and non-convergent linear methods as preconditioners for conjugate gradient methods, and it is applied to domain decomposition and multigrid. This paper illustrates why symmetrizing may be a bad idea for linear methods. Numerical examples are presented for a test problem.

INTRODUCTION

In this paper, we consider additive and multiplicative Schwarz methods and their acceleration with Krylov methods, for the numerical solution of self-adjoint positive definite (SPD) operator equations arising from the discretization of elliptic partial differential equations. The standard theory of conjugate gradient acceleration of linear methods requires that a certain operator associated with the linear method—the preconditioner—be symmetric and positive definite. Often, however, as in the case of Schwarz-based preconditioners, the preconditioner is known only implicitly, and symmetry and positive definiteness are not easily verified. Here, we try to construct natural sets of sufficient conditions that are easily verified and do not require the explicit formulation of the preconditioner. More precisely, we derive conditions for the constituent components of MG and DD algorithms (smoother, subdomain solver, transfer operators, etc.), that guarantee symmetry and positive definiteness of the preconditioning operator which is (explicitly or implicitly) defined by the resulting Schwarz method. We examine the implications of these conditions for various formulations of the standard DD and MG algorithms.

The outline of the paper is as follows. We begin in the next section by reviewing basic linear methods for SPD linear operator equations and by examining Krylov acceleration strategies. A simple lemma will illustrate why symmetrizing may be a bad idea for linear methods. In the third and fourth sections, we analyze multiplicative and additive Schwarz preconditioners. We develop a theory that establishes sufficient conditions for the multiplicative and additive algorithms to yield SPD preconditioners. This theory is used to establish sufficient conditions for multiplicative and additive DD and MG methods, and it allows for analysis of non-variational and even non-convergent linear methods as preconditioners. In the final section, we report results of numerical experiments with finite-element-based DD and MG methods applied to a difficult test problem with discontinuous coefficients to illustrate the theory and conjectures.

[1]This work was supported in part by the NSF under Cooperative Agreement No. CCR-9120008.

Notation. Let $\mathcal{H}$ be a real finite-dimensional Hilbert space equipped with the inner-product $(\cdot,\cdot)$ inducing the norm $\|\cdot\| = (\cdot,\cdot)^{1/2}$. The *adjoint* of a linear operator $A \in \mathbf{L}(\mathcal{H},\mathcal{H})$ with respect to $(\cdot,\cdot)$ is the unique operator A^T satisfying $(Au,v) = (u,A^Tv)$, $\forall u,v \in \mathcal{H}$. An operator A is called *self-adjoint* or *symmetric* if $A = A^T$; a self-adjoint operator A is called *positive definite* or simply *positive* if $(Au,u) > 0$, $\forall u \in \mathcal{H}$, $u \neq 0$. If A is self-adjoint positive definite (SPD), then the bilinear form (Au,v) defines another inner-product, which we denote as $(\cdot,\cdot)_A$. It induces the norm $\|\cdot\|_A = (\cdot,\cdot)_A^{1/2}$.

The adjoint of an operator M with respect to $(\cdot,\cdot)_A$, the *A-adjoint*, is the unique operator M^* satisfying $(Mu,v)_A = (u,M^*v)_A$, $\forall u,v \in \mathcal{H}$. From this definition it follows that

$$M^* = A^{-1}M^TA\ . \tag{1}$$

M is called *A-self-adjoint* if $M = M^*$ and *A-positive* if $(Mu,u)_A > 0$, $\forall u \in \mathcal{H}$, $u \neq 0$.

If $N \in \mathbf{L}(\mathcal{H}_1,\mathcal{H}_2)$, then $N^T \in \mathbf{L}(\mathcal{H}_2,\mathcal{H}_1)$ is defined as the unique operator relating the inner-products in $\mathcal{H}_1$ and $\mathcal{H}_2$ as follows:

$$(Nu,v)_{\mathcal{H}_2} = (u,N^Tv)_{\mathcal{H}_1}\ , \quad \forall u \in \mathcal{H}_1\ , \quad \forall v \in \mathcal{H}_2\ . \tag{2}$$

Since it is usually clear from the arguments which inner-product is involved, we shall often drop the subscripts on inner-products (and norms) throughout the paper.

We denote the spectrum of an operator M as $\sigma(M)$. The spectral theory for self-adjoint linear operators states that the eigenvalues of the self-adjoint operator M are real and lie in the closed interval $[\lambda_{\min}(M),\lambda_{\max}(M)]$ defined by the Raleigh quotients:

$$\lambda_{\min}(M) = \min_{u\neq 0}\frac{(Mu,u)}{(u,u)}, \qquad \lambda_{\max}(M) = \max_{u\neq 0}\frac{(Mu,u)}{(u,u)}.$$

Similarly, if an operator M is A-self-adjoint, then its eigenvalues are real and lie in the interval defined by the Raleigh quotients generated by the A-inner-product. A well-known property is that if M is self-adjoint, then the spectral radius of M, denoted as $\rho(M)$, satisfies $\rho(M) = \|M\|$. This property can also be shown to hold in the A-norm for A-self-adjoint operators.

Lemma 1. *If A is SPD and M is A-self-adjoint, then $\rho(M) = \|M\|_A$.*

Linear methods. Given the equation $Au = f$, where $A \in \mathbf{L}(\mathcal{H},\mathcal{H})$ is SPD, consider the *preconditioned* equation $BAu = Bf$, with $B \in \mathbf{L}(\mathcal{H},\mathcal{H})$. The operator B, the *preconditioner*, is usually chosen so that the linear iteration

$$u^{n+1} = u^n - BAu^n + Bf = (I - BA)u^n + Bf, \tag{3}$$

has some desired convergence properties. The convergence of (3) is determined by the properties of the so-called *error propagation operator*, $E = I - BA$.

We now state a series of simple lemmas that we shall use repeatedly in the following sections. Their short proofs and further references can be found in [5].

Lemma 2. *If A is SPD, then BA is A-self-adjoint if and only if B is self-adjoint.*

Lemma 3. *If A is SPD, then E is A-self-adjoint if and only if B is self-adjoint.*

Lemma 4. *If A and B are SPD, then BA is A-SPD.*

Lemma 5. *If A is SPD and B is self-adjoint, then $\|E\|_A = \rho(E)$.*

Lemma 6. *If E^* is the A-adjoint of E, then $\|E\|_A^2 = \|EE^*\|_A$.*

Lemma 7. *If A and B are SPD and E is A-non-negative, then $\|E\|_A < 1$.*

Lemma 8. *If A is SPD and B is self-adjoint, and E is such that*

$$-C_1(u,u)_A \leq (Eu,u)_A \leq C_2(u,u)_A, \qquad \forall u \in \mathcal{H},$$

for $C_1 \geq 0$ and $C_2 \geq 0$, then $\rho(E) = \|E\|_A \leq \max\{C_1, C_2\}$.

Lemma 9. *If A and B are SPD, then Lemma 8 holds for some $C_2 < 1$.*

The following lemma illustrates why symmetrizing is a bad idea for linear methods. It exposes the convergence rate penalty incurred by symmetrization of a linear method.

Lemma 10. *For any $E \in \mathbf{L}(\mathcal{H}, \mathcal{H})$, it holds that:*

$$\rho(EE) \leq \|EE\|_A \leq \|E\|_A^2 = \|EE^*\|_A = \rho(EE^*).$$

Proof. The first and second inequalities hold for any norm. The first equality follows from Lemma 6, and the second follows from Lemma 1. □

Note that this is an inequality not only for the spectral radii but also for the A-norms of the nonsymmetric and symmetrized error propagators. The lemma illustrates that one may actually see the differing convergence rates early in the iteration as well.

Krylov acceleration of SPD linear methods. The conjugate gradient method was developed by Hestenes and Stiefel [4] as a method for solving linear systems $Au = f$, with SPD operators A. In order to improve convergence, it is common to *precondition* the linear system by an SPD *preconditioning operator* $B \approx A^{-1}$, in which case the generalized or preconditioned conjugate gradient method results. Our goal in this section is to briefly review some relationships between the contraction number of a basic linear preconditioner and that of the resulting preconditioned conjugate gradient algorithm.

We start with the well-known conjugate gradient contraction bound [3]

$$\|e^{i+1}\|_A \;\leq\; 2\left(1 - \frac{2}{1+\sqrt{\kappa_A(BA)}}\right)^{i+1} \|e^0\|_A \;=\; 2\,\delta_{\mathrm{cg}}^{i+1}\,\|e^0\|_A,$$

where $\kappa_A(BA)$, the A-condition number of BA, is the ratio of extreme eigenvalues of BA.

The following result gives a bound on the condition number of the operator BA in terms of the extreme eigenvalues of the error propagator $E = I - BA$; such bounds are often used in the analysis of linear preconditioners (cf. Proposition 5.1 in [9]).

Lemma 11. *If A and B are SPD and E is such that*

$$-C_1(u,u)_A \leq (Eu,u)_A \leq C_2(u,u)_A, \qquad \forall u \in \mathcal{H},$$

for $C_1 \geq 0$ and $C_2 \geq 0$, then the above must hold with $C_2 < 1$, and it follows that

$$\kappa_A(BA) \leq \frac{1+C_1}{1-C_2}.$$

Remark 1. Even if a linear method is not convergent, it may still be a good preconditioner. If $C_2 << 1$ and if $C_1 > 1$ does not become too large, then $\kappa_A(BA)$ will be small and the conjugate gradient method will converge rapidly, even though the linear method diverges.

The next result connects the contraction number of the preconditioner to the contraction number of the preconditioned conjugate gradient method (see [10] for a proof).

Lemma 12. *If A and B are SPD and $\|I - BA\|_A \leq \delta < 1$, then $\delta_{\text{cg}} < \delta$.*

Krylov acceleration of nonsymmetric linear methods. The convergence theory of the conjugate gradient iteration requires that the preconditioned operator BA be A-self-adjoint (see [1] for more general conditions), which from Lemma 2 requires that B be self-adjoint. If a Schwarz method is employed which produces a nonsymmetric operator B, then although A is SPD, the theory of the previous section does not apply and a nonsymmetric solver such as conjugate gradients on the normal equations [1], GMRES [6], CGS [7], or Bi-CGstab [8] must be used. Further on, we shall use the preconditioned Bi-CGstab algorithm to accelerate nonsymmetric Schwarz methods. In a sequence of numerical experiments, we shall compare the effectiveness of this approach with unaccelerated symmetric and nonsymmetric Schwarz methods, and with symmetric Schwarz methods accelerated with conjugate gradients.

MULTIPLICATIVE SCHWARZ METHODS

Consider a product operator of the form:

$$E = I - BA = (I - \bar{B}_1 A)(I - B_0 A)(I - B_1 A), \tag{4}$$

where $\bar{B}_1, B_0$, and B_1 are linear operators on $\mathcal{H}$, and where A is, as before, an SPD operator on $\mathcal{H}$. We are interested in conditions for $\bar{B}_1, B_0$, and B_1, which guarantee that the implicitly defined operator B is self-adjoint and positive definite and, hence, can be accelerated by using the conjugate gradient method.

Lemma 13. *Sufficient conditions for symmetry and positivity of operator B, defined by (4), are:*

1. $\bar{B}_1 = B_1^T$;
2. $B_0 = B_0^T$;
3. $\|I - B_1 A\|_A < 1$;

4. B_0 non-negative on $\mathcal{H}$.

Proof. By Lemma 3, in order to prove symmetry of B, it is sufficient to prove that E is A-self-adjoint. By using (1), we get

$$E^* = A^{-1}E^TA = (I - B_1^TA)(I - B_0^TA)(I - \bar{B}_1^TA),$$

which equals E following from conditions 1 and 2.

Next, we prove that $(Bu, u) > 0$, $\forall u \in \mathcal{H}$, $u \neq 0$. Since A is non-singular, this is equivalent to proving that $(BAu, Au) > 0$. Using condition 1, we have that

$$\begin{aligned}(BAu, Au) &= ((I - E)u, Au) \\ &= (u, Au) - ((I - B_1^TA)(I - B_0A)(I - B_1A)u, Au) \\ &= (u, Au) - ((I - B_0A)(I - B_1A)u, A(I - B_1A)u) \\ &= (u, Au) - ((I - B_1A)u, A(I - B_1A)u) + (B_0w, w),\end{aligned}$$

where $w = A(I - B_1A)u$. By condition 4, we have that $(B_0w, w) \geq 0$. Condition 3 implies that $((I - B_1A)u, A(I - B_1A)u) < (u, Au)$ for $u \neq 0$. Thus, the first two terms in the sum above are together positive, while the third is non-negative, so that B is positive. □

Multiplicative domain decomposition. Given the finite-dimensional Hilbert space $\mathcal{H}$, consider J spaces $\mathcal{H}_k$, $k = 1, \ldots, J$, together with linear operators $I_k \in \mathbf{L}(\mathcal{H}_k, \mathcal{H})$, $\text{null}(I_k) = \{0\}$, such that $I_k\mathcal{H}_k \subseteq \mathcal{H} = \sum_{k=1}^J I_k\mathcal{H}_k$. We also assume the existence of another space $\mathcal{H}_0$, an associated operator I_0 such that $I_0\mathcal{H}_0 \subseteq \mathcal{H}$, and some linear operators $I^k \in \mathbf{L}(\mathcal{H}, \mathcal{H}_k), k = 0, \ldots, J$. For notational convenience, we shall denote the inner-products on $\mathcal{H}_k$ by $(\cdot, \cdot)$ (without explicit reference to the particular space). Note that the inner products on different spaces need not be related.

In a domain decomposition context, the spaces $\mathcal{H}_k$, $k = 1, \ldots, J$ are typically associated with *local subdomains* of the original domain on which the partial differential equation is defined. The space $\mathcal{H}_0$ is then a space associated with some global coarse mesh. The operators $I_k, k = 1, \ldots, J$ are usually inclusion operators, while I_0 is an interpolation or prolongation operator (as in a two-level MG method). The operators $I^k, k = 1, \ldots, J$ are usually orthogonal projection operators, while I^0 is a restriction operator (again, as in a two-level MG method).

The error propagator of a multiplicative DD method on the space $\mathcal{H}$ employing the subspaces $I_k\mathcal{H}_k$ has the general form [2]

$$E = I - BA = (I - I_J\bar{R}_JI^JA) \cdots (I - I_0R_0I^0A) \cdots (I - I_JR_JI^JA)\ , \tag{5}$$

where $\bar{R}_k$ and R_k, $k = 1, \ldots, J$, are linear operators on $\mathcal{H}_k$ and R_0 is a linear operator on $\mathcal{H}_0$. Usually the operators $\bar{R}_k$ and R_k are constructed so that $\bar{R}_k \approx A_k^{-1}$ and $R_k \approx A_k^{-1}$, where A_k is the operator defining the subdomain problem in $\mathcal{H}_k$. Similarly, R_0 is constructed so that $R_0 \approx A_0^{-1}$. Actually, quite often R_0 is a "direct solve", i.e., $R_0 = A_0^{-1}$. The subdomain problem operator A_k is related to the restriction of A to $\mathcal{H}_k$. We say that A_k satisfies the *Galerkin conditions* or, in a finite element setting, that it is *variationally* defined when

$$A_k = I^kAI_k, \qquad I^k = I_k^T. \tag{6}$$

Recall that the superscript "T" is to be interpreted as the adjoint in the sense of (2), i.e., with respect to the inner-products in $\mathcal{H}$ and $\mathcal{H}_k$.

Propagator (5) can be thought of as the product operator (4) by choosing

$$I - \bar{B}_1 A = \prod_{k=J}^{1} (I - I_k \bar{R}_k I^k A) \ , \quad B_0 = I_0 R_0 I^0 \ , \quad I - B_1 A = \prod_{k=1}^{J} (I - I_k R_k I^k A) \ ,$$

where $\bar{B}_1$ and B_1 are known only implicitly. This identification allows for the use of Lemma 13 to establish sufficient conditions on the subdomain operators $\bar{R}_k$, R_k, and R_0 to guarantee that multiplicative domain decomposition yields an SPD operator B.

Theorem 1. *Sufficient conditions for symmetry and positivity of the multiplicative domain decomposition operator B, defined by (5), are:*

1. $I^k = c_k I_k^T \ , \quad c_k > 0 \ , \quad k = 0, \cdots, J \ ;$
2. $\bar{R}_k = R_k^T \ , \quad k = 1, \cdots, J \ ;$
3. $R_0 = R_0^T \ ;$
4. $\left\| \prod_{k=1}^{J} (I - I_k R_k I^k A) \right\|_A < 1 \ ;$
5. R_0 *non-negative on* $\mathcal{H}_0$.

Proof. We show that the conditions of Lemma 13 are satisfied. First, we prove that $\bar{B}_1 = B_1^T$, which, by Lemma 3, is equivalent to proving that $(I - B_1 A)^* = (I - \bar{B}_1 A)$. By using (1), we have

$$\left(\prod_{k=1}^{J} (I - I_k R_k I^k A) \right)^* = A^{-1} \left(\prod_{k=1}^{J} (I - I_k R_k I^k A) \right)^T A = \prod_{k=J}^{1} (I - (I^k)^T R_k^T (I_k)^T A) \ ,$$

which equals $(I - \bar{B}_1 A)$ under conditions 1 and 2 of the theorem. The symmetry of B_0 follows immediately from conditions 1 and 3; indeed,

$$B_0^T = (I_0 R_0 I^0)^T = (I^0)^T R_0^T (I_0)^T = (c_0 I_0) R_0 (c_0^{-1} I^0) = I_0 R_0 I^0 = B_0 \ .$$

By condition 4 of the theorem, condition 3 of Lemma 13 holds trivially. The theorem follows if one realizes that condition 4 of Lemma 13 is also satisfied, since,

$$(B_0 u, u) = (I_0 R_0 I^0 u, u) = (R_0 I^0 u, I_0^T u) = c_0^{-1} (R_0 I^0 u, I^0 u) \geq 0 \ , \quad \forall u \in \mathcal{H} \ .$$

□

Remark 2. Note that one sweep through the subdomains, followed by a coarse problem solve, followed by another sweep through the subdomains in reverse order, gives rise to an error propagator of the form (5). Also, note that no conditions are imposed on the nature of the operators A_k associated with each subdomain. In particular, the theorem *does not* require that the variational conditions be satisfied. The theorem also does not require that the overall multiplicative DD method be convergent.

Remark 3. The results of the theorem apply for operators on general finite-dimensional Hilbert spaces with arbitrary inner-products. They hold in particular for matrix operators on $\mathbb{R}^N$, equipped with the Euclidean inner-product or the discrete L_2 inner-product. In the former case the superscript "T" corresponds to the standard matrix transpose. In the latter case, the matrix representation of the adjoint is a scalar multiple of the matrix transpose; the scalar may be different from unity when the adjoint involves two different spaces, as in the case of prolongation and restriction. This possible constant in the case of the discrete L_2 inner-product is absorbed in the factor c_k in condition 1. This allows for an easy verification of the conditions of the theorem in an actual implementation, where the operators are represented as matrices and where the inner-products do not explicitly appear in the algorithm.

Remark 4. Condition 1 of the theorem (with $c_k = 1$) for $k = 1, \ldots, J$ is usually satisfied trivially for domain decomposition methods. For $k = 0$, it may have to be imposed explicitly. Condition 2 of the theorem allows for several alternatives which give rise to an SPD preconditioner, namely: (1) use of exact subdomain solvers (if A_k is a symmetric operator); (2) use of identical symmetric subdomain solvers in the forward and backward sweeps; and (3) use of the adjoint of the subdomain solver on the second sweep. Condition 3 is satisfied when the coarse problem is symmetric and the solve is an exact one, which is usually the case. If not, the coarse problem solve has to be symmetric. Condition 4 in Theorem 1 is clearly a non-trivial one; it is essentially the assumption that the multiplicative DD method without a coarse space is convergent. Condition 5 is satisfied, for example, when the coarse problem is SPD and the solve is exact.

Multiplicative multigrid. Consider the Hilbert space $\mathcal{H}$ and J spaces $\mathcal{H}_k$ together with operators $I_k \in \mathbf{L}(\mathcal{H}_k, \mathcal{H})$, $\text{null}(I_k) = 0$, such that the spaces $I_k\mathcal{H}_k$ are nested and satisfy $I_1\mathcal{H}_1 \subseteq I_2\mathcal{H}_2 \subseteq \cdots \subseteq I_{J-1}\mathcal{H}_{J-1} \subseteq \mathcal{H}_J \equiv \mathcal{H}$. As before, we denote the $\mathcal{H}_k$-inner-products by $(\cdot, \cdot)$, since it will be clear from the arguments which inner-product is intended. Again, the inner-products are not necessarily related in any way. We assume also the existence of operators $I^k \in \mathbf{L}(\mathcal{H}, \mathcal{H}_k)$.

In a multigrid context, the spaces $\mathcal{H}_k$ are typically associated with a nested hierarchy of successively refined meshes, with $\mathcal{H}_1$ being the coarsest mesh and $\mathcal{H}_J$ being the fine mesh on which the PDE solution is desired. The linear operators I_k are prolongation operators, constructed from given interpolation or prolongation operators that operate between subspaces, i.e., $I^k_{k-1} \in \mathbf{L}(\mathcal{H}_{k-1}, \mathcal{H}_k)$. The operator I_k is then constructed (only as a theoretical tool) as a composite operator

$$I_k = I^J_{J-1} I^{J-1}_{J-2} \cdots I^{k+2}_{k+1} I^{k+1}_k, \quad k = 1, \ldots, J-1. \tag{7}$$

The composite restriction operators I^k, $k = 1, \ldots, J-1$, are constructed similarly from some given restriction operators $I^{k-1}_k \in \mathbf{L}(\mathcal{H}_k, \mathcal{H}_{k-1})$. The coarse problem operators A_k are related to the restriction of A to $\mathcal{H}_k$. As in the case of DD methods, we say that A_k is *variationally* defined or satisfies the *Galerkin conditions* when conditions (6) hold. It is not difficult to see that conditions (6) are equivalent to the following recursively defined variational conditions:

$$A_k = I^k_{k+1} A_{k+1} I^{k+1}_k, \qquad I^k_{k+1} = (I^{k+1}_k)^T. \tag{8}$$

when the composite operators I_k appearing in (6) are defined as in (7).

In a finite element setting, conditions (8) can be shown to hold in ideal situations, for both the stiffness matrices and the abstract weak form operators, for a nested sequence of successively refined finite element meshes. In the finite difference or finite volume method setting, conditions (8) must often be imposed algebraically, in a recursive fashion.

The error propagator of a multiplicative V-cycle MG method is defined implicitly as

$$E = I - BA = I - D_J A_J, \tag{9}$$

where $A_J = A$ and where operators D_k, $k = 2, \ldots, J$, are defined recursively:

$$I - D_k A_k = (I - \bar{R}_k A_k)(I - I_{k-1}^k D_{k-1} I_k^{k-1} A_k)(I - R_k A_k), \; k = 2, \ldots, J, \tag{10}$$

$$D_1 = R_1 \; . \tag{11}$$

Operators $\bar{R}_k$ and R_k are linear operators on $\mathcal{H}_k$, usually called *smoothers*. The linear operators $A_k \in L(\mathcal{H}_k, \mathcal{H}_k)$ define the coarse problems. They often satisfy the variational condition (8).

The error propagator (9) can be thought of as an operator of the form (4) with

$$\bar{B}_1 = \bar{R}_J \; , \quad B_0 = I_{J-1}^J D_{J-1} I_J^{J-1} \; , \quad B_1 = R_J \; .$$

Such an identification with the product method allows for the use of Lemma 13. The following theorem establishes sufficient conditions for the subspace operators R_k, $\bar{R}_k$, and A_k in order to generate an (implicitly defined) SPD operator B that can be accelerated with conjugate gradients.

Theorem 2. *Sufficient conditions for symmetry and positivity of the multiplicative multigrid operator B, implicitly defined by (9), (10), and (11), are*

1. *A_k is SPD on $\mathcal{H}_k$, $k = 2, \ldots, J$;*
2. *$I_k^{k-1} = c_k (I_{k-1}^k)^T$, $c_k > 0$, $k = 2, \ldots, J$;*
3. *$\bar{R}_k = R_k^T$, $k = 2, \ldots, J$;*
4. *$R_1 = R_1^T$;*
5. *$\|I - R_J A\|_A < 1$;*
6. *$\|I - R_k A_k\|_{A_k} \leq 1$, $k = 2, \ldots, J-1$;*
7. *R_1 non-negative on $\mathcal{H}_1$.*

Proof. Since $\bar{R}_J = R_J^T$, we have that $\bar{B}_1 = B_1^T$, which gives condition 1 of Lemma 13. Now, B_0 is symmetric if and only if

$$B_0 = I_{J-1}^J D_{J-1} I_J^{J-1} = (c_J^{-1} I_J^{J-1})^T D_{J-1}^T (c_J I_{J-1}^J)^T = B_0^T,$$

which holds under condition 2 and a symmetry requirement for D_{J-1}. We will prove that $D_{J-1} = D_{J-1}^T$ by induction. First, $D_1 = D_1^T$ since $R_1 = R_1^T$. By Lemma 3 and condition 1, D_k is symmetric if and only if $E_k = I - D_k A_k$ is A_k-self-adjoint. By using (1), we have that

$$\begin{aligned} E_k^* &= A_k^{-1} \left((I - \bar{R}_k A_k)(I - I_{k-1}^k D_{k-1} I_k^{k-1} A_k)(I - R_k A_k) \right)^T A_k \\ &= (I - \bar{R}_k A_k)(I - (c_k I_{k-1}^k) D_{k-1}^T (c_k^{-1} I_k^{k-1}) A_k)(I - R_k A_k), \end{aligned}$$

where we have used conditions 1, 2, and 3. Therefore, $E_k^* = E_k$, if $D_{k-1} = D_{k-1}^T$. Hence, the result follows by induction on k.

Condition 3 of Lemma 13 follows trivially by condition 5 of the theorem.

It remains to verify condition 4 of Lemma 13, namely that B_0 is non-negative. This is equivalent to showing that D_{J-1} is non-negative on $\mathcal{H}_{J-1}$. This will follow again from an induction argument. First, note that $D_1 = R_1$ is non-negative on $\mathcal{H}_1$. Next, we prove that $(D_k v_k, v_k) \geq 0$, $\forall v_k \in \mathcal{H}_k$, or, equivalently, since A_k is non-singular, that $(D_k A_k v_k, A_k v_k) \geq 0$. So, for all $v_k \in \mathcal{H}_k$,

$$\begin{aligned}
(D_k A_k v_k, A_k v_k) &= (A_k v_k, v_k) - (A_k E_k v_k, v_k) \\
&= (A_k v_k, v_k) - (A_k (I - I_{k-1}^k D_{k-1} I_k^{k-1} A_k)(I - R_k A_k) v_k, (I - R_k A_k) v_k) \\
&= (A_k v_k, v_k) - (A_k (I - R_k A_k) v_k, (I - R_k A_k) v_k) \\
&\quad + (A_k I_{k-1}^k D_{k-1} I_k^{k-1} A_k (I - R_k A_k) v_k, (I - R_k A_k) v_k) \\
&= (v_k, v_k)_{A_k} - (S_k v_k, S_k v_k)_{A_k} + c_k^{-1} (D_{k-1} v_{k-1}, v_{k-1})
\end{aligned}$$

where $S_k = I - R_k A_k$ and $v_{k-1} = I_k^{k-1} A_k (I - R_k A_k) v_k \in \mathcal{H}_{k-1}$. By condition 6, the first two terms add up to a non-negative value. Hence, D_k is non-negative if D_{k-1} is non-negative. □

Remark 5. As noted earlier in Remark 3, the conditions and conclusions of the theorem can be interpreted completely in terms of the usual matrix representations of the multigrid operators.

Remark 6. Condition 1 of the theorem requires all but the coarsest grid operator to be SPD. This is easily satisfied when they are constructed either by discretization or by explicitly enforcing the Galerkin condition. Condition 2 requires restriction and prolongation to be adjoints, possibly multiplied by an arbitrary constant. Condition 3 of the theorem is satisfied when the number of pre-smoothing steps equals the number of post-smoothing steps and, in addition, one of the following is imposed: (1) use of the same symmetric smoother for both pre- and post-smoothing; or (2) use of the adjoint of the pre-smoothing operator as the post-smoother. Condition 4 requires a symmetric coarsest mesh solver. When the coarsest mesh problem is SPD, the symmetry of R_1 is satisfied when it corresponds to an exact solve (as is typical for MG methods). Condition 5 is a convergence requirement on the fine space smoother. Condition 6 requires the coarse grid smoothers to be non-divergent. The non-negativity requirement for R_1 is a non-trivial one; however, if A_1 is SPD, it is immediately satisfied when the operator corresponds to an exact solve.

ADDITIVE SCHWARZ METHODS

Consider a sum operator of the following form:

$$E = I - BA = I - \omega(B_0 + B_1)A, \quad \omega > 0, \tag{12}$$

where, as before, A is an SPD operator and B_0 and B_1 are linear operators on $\mathcal{H}$.

Lemma 14. *Sufficient conditions for symmetry and positivity of B, defined in (12), are*

1. B_1 *is SPD in* $\mathcal{H}$;

2. B_0 *is symmetric and non-negative on* $\mathcal{H}$.

Proof. We have that $B = \omega(B_0 + B_1)$, which is symmetric by the symmetry of B_0 and B_1. Positivity follows since $(B_0 u, u) \geq 0$ and $(B_1 u, u) > 0$, $\forall u \in \mathcal{H}$, $u \neq 0$. □

Additive domain decomposition. We consider the space $\mathcal{H}$ and the J subspaces $I_k\mathcal{H}_k$ such that $I_k\mathcal{H}_k \subseteq \mathcal{H} = \sum_{k=1}^{J} I_k\mathcal{H}_k$. Again, we allow for a "coarse" subspace $I_0\mathcal{H}_0 \subseteq \mathcal{H}$.

The error propagator of an additive DD method on the space $\mathcal{H}$ employing the subspaces $I_k\mathcal{H}_k$ has the general form (see [10])

$$E = I - BA = I - \omega(I_0 R_0 I^0 + I_1 R_1 I^1 + \cdots + I_J k_J I^J)A. \tag{13}$$

The operators R_k are constructed in such a way that $R_k \approx A_k^{-1}$, where the A_k are the subdomain problem operators. Propagator (13) can be thought of as the sum method (12) by taking $B_0 = I_0 R_0 I^0$ and $B_1 = \sum_{k=1}^{J} I_k R_k I^k$. This identification allows for the use of Lemma 14 in order to establish conditions to guarantee that additive domain decomposition yields an SPD preconditioner. Before we state the main theorem, we need the following lemma, which characterizes the splitting of $\mathcal{H}$ into subspaces $I_k\mathcal{H}_k$ in terms of a positive *splitting constant* S_0.

Lemma 15. *Given any* $v \in \mathcal{H}$, *there exists a splitting* $v = \sum_{k=1}^{J} I_k v_k$, $v_k \in \mathcal{H}_k$, *and a constant* $S_0 > 0$ *such that*

$$\sum_{k=1}^{J} \|I_k v_k\|_A^2 \leq S_0 \|v\|_A^2. \tag{14}$$

Proof. Since $\sum_{k=1}^{J} I_k\mathcal{H}_k = \mathcal{H}$, we can construct subspaces $\mathcal{V}_k \subseteq \mathcal{H}_k$ such that $I_k\mathcal{V}_k \cap I_l\mathcal{V}_l = \{0\}$, for $k \neq l$ and $\mathcal{H} = \sum_{k=1}^{J} I_k\mathcal{V}_k$. Any $v \in \mathcal{H}$, can be decomposed uniquely as $v = \sum_{k=1}^{J} I_k v_k$, $v_k \in \mathcal{V}_k$. Define the projectors $Q_k \in L(\mathcal{H}, I_k\mathcal{V}_k)$ such that $Q_k v = I_k v_k$. Then,

$$\sum_{k=1}^{J} \|I_k v_k\|_A^2 = \sum_{k=1}^{J} \|Q_k v\|_A^2 \leq \sum_{k=1}^{J} \|Q_k\|_A^2 \, \|v\|_A^2.$$

Hence, the result follows with $S_0 = \sum_{k=1}^{J} \|Q_k\|_A^2$. □

Theorem 3. *Sufficient conditions for symmetry and positivity of the additive domain decomposition operator* B, *defined in (13), are*

1. $I^k = c_k I_k^T$, $\quad c_k > 0, \quad k = 0, \ldots, J$;

2. R_k *is SPD on* $\mathcal{H}_k$, $\quad k = 1, \ldots, J$;

3. R_0 *is symmetric and non-negative on* $\mathcal{H}_0$.

Proof. Symmetry of B_0 and B_1 follow trivially from the symmetry of R_k and R_0 and from $I^k = c_k I_k^T$. That B_0 is non-negative on $\mathcal{H}$ follows immediately from the non-negativity of R_0 on $\mathcal{H}_0$.

Finally, we prove positivity of B_1. Define $A_k = I^k A I_k$, $k = 1, \ldots, J$. By condition 1 and the full rank nature of I_k, we have that A_k is SPD. Now, since R_k is also SPD, the product $R_k A_k$ is A_k-SPD. Hence, there exists an $\omega_0 > 0$ such that $0 < \omega_0 < \lambda_i(R_k A_k)$, $k = 1, \ldots, J$. This is used together with (14) to bound the sum

$$\sum_{k=1}^{J} c_k^{-1}(R_k^{-1}v_k, v_k) = \sum_{k=1}^{J} c_k^{-1}(A_k A_k^{-1} R_k^{-1} v_k, v_k) \leq \sum_{k=1}^{J} c_k^{-1}(A_k v_k, v_k) \max_{v_k \neq 0} \frac{(A_k A_k^{-1} R_k^{-1} v_k, v_k)}{(A_k v_k, v_k)}$$

$$\leq \sum_{k=1}^{J} c_k^{-1}\omega_0^{-1}(A_k v_k, v_k) = \sum_{k=1}^{J} \omega_0^{-1}(A I_k v_k, I_k v_k) = \sum_{k=1}^{J} \omega_0^{-1} \|I_k v_k\|_A^2 \leq \left(\frac{S_0}{\omega_0}\right) \|v\|_A^2,$$

with $v = \sum_{k=1}^{J} I_k v_k$. We can now employ this result to establish positivity of B_1:

$$\|v\|_A^2 = (Av, v) = \sum_{k=1}^{J} (Av, I_k v_k) = \sum_{k=1}^{J} (I_k^T A v, v_k) = \sum_{k=1}^{J} (R_k c_k^{1/2} I_k^T A v, R_k^{-1} c_k^{-1/2} v_k).$$

By using the Cauchy-Schwarz inequality first in the R_k-inner-product and then in $\mathbb{R}^J$, we have that

$$\|v\|_A^2 \leq \left(\sum_{k=1}^{J} (R_k R_k^{-1} c_k^{-1/2} v_k, R_k^{-1} c_k^{-1/2} v_k)\right)^{1/2} \left(\sum_{k=1}^{J} (R_k c_k^{1/2} I_k^T A v, c_k^{1/2} I_k^T A v)\right)^{1/2}$$

$$\leq \left(\frac{S_0}{\omega_0}\right)^{1/2} \|v\|_A \left(\sum_{k=1}^{J} (I_k R_k c_k I_k^T A v, Av)\right)^{1/2} = \left(\frac{S_0}{\omega_0}\right)^{1/2} \|v\|_A \, (B_1 Av, Av)^{1/2} .$$

Finally, we divide by $\|v\|_A$ and square to obtain

$$(B_1 Av, Av) \geq \frac{\omega_0}{S_0} \|v\|_A^2 > 0 , \quad \forall v \in \mathcal{H}, \quad v \neq 0.$$

□

Remark 7. Condition 1 is naturally satisfied for $k = 1, \ldots, J$, with $c_k = 1$, since the associated I_k and I^k are usually inclusion and orthogonal projection operators (which are natural adjoints when the inner-products are inherited from the parent space, as in domain decomposition). The fact that $I^0 = c_0 I_0^T$ needs to be established explicitly. Condition 2 requires the use of SPD subdomain solvers. The condition will hold, for example, when the subdomain solve is exact and the subdomain problem operator is SPD. (The latter is naturally satisfied by condition 1 and the full rank nature of I_k.) Finally, condition 3 is nontrivial and needs to be checked explicitly. The condition holds when the coarse space problem operator is SPD and the solve is exact. Note that variational conditions are not needed for the coarse space problem operator.

Additive multigrid. Given are the Hilbert space $\mathcal{H}$ and $J-1$ nested subspaces $I_k \mathcal{H}_k$ such that $I_1 \mathcal{H}_1 \subseteq I_2 \mathcal{H}_2 \subseteq \cdots \subseteq I_{J-1} \mathcal{H}_{J-1} \subseteq \mathcal{H}_J \equiv \mathcal{H}$. The operators I_k and I^k are the usual linear operators between the different spaces, as in the previous sections.

The error propagator of an additive MG method is defined explicitly:

$$E = I - BA = I - \omega(I_1 R_1 I^1 + I_2 R_2 I^2 + \cdots + I_{J-1} R_{J-1} I^{J-1} + R_J)A. \tag{15}$$

This can be thought of as the sum method analyzed earlier by taking $B_0 = \sum_{k=1}^{J-1} I_k R_k I^k$ and $B_1 = R_J$. This identification allows for the use of Lemma 14 to establish sufficient conditions to guarantee that additive MG yields an SPD preconditioner.

Theorem 4. *Sufficient conditions for symmetry and positivity of the additive multigrid operator B defined in (15) are*

1. $I^k = c_k I_k^T, \quad c_k > 0, \quad k = 1, \ldots, J-1;$
2. R_J *is SPD in* $\mathcal{H}$;
3. R_k *is symmetric non-negative in* $\mathcal{H}_k, \quad k = 1, \ldots, J-1.$

Proof. Symmetry of B_0 and B_1 is obvious. B_1 is positive by condition 2. Non-negativity of B_0 follows from

$$(B_0 u, u) = \sum_{k=1}^{J-1} (I_k R_k (c_k I_k)^T u, u) = \sum_{k=1}^{J-1} c_k (R_k I_k^T u, I_k^T u) \geq 0, \quad \forall u \in \mathcal{H}, u \neq 0.$$

□

Remark 8. Condition 1 of the theorem has to be imposed explicitly. Conditions 2 and 3 require the smoothers to be symmetric. The positivity of R_J is satisfied when the fine grid smoother is convergent, although this is not a necessary condition. The non-negativity of R_k, $k < J$, has to be checked explicitly. When the coarse problem operators A_k are SPD, this condition is satisfied, for example, when the smoothers are non-divergent. Note that variational conditions for the subspace problem operators are not required.

NUMERICAL RESULTS

The Poisson-Boltzmann equation describes the electrostatic potential of a biomolecule lying in an ionic solvent. This nonlinear elliptic equation for the dimensionless electrostatic potential $u(\mathbf{r})$ has the form

$$-\nabla \cdot (\epsilon(\mathbf{r}) \nabla u(\mathbf{r})) + \bar{\kappa}^2 \sinh(u(\mathbf{r})) = \left(\frac{4\pi e_c^2}{k_B T}\right) \sum_{i=1}^{N_m} z_i \delta(\mathbf{r} - \mathbf{r}_i), \quad \mathbf{r} \in \mathbb{R}^3, \qquad u(\infty) = 0.$$

The coefficients appearing in the equation are discontinuous by orders of magnitude. The placement and magnitude of atomic charges are represented by source terms involving delta-functions. Analytical techniques are used to obtain boundary conditions on a finite domain boundary.

We will compare several MG and DD methods for a two-dimensional, linearized form of the Poisson-Boltzmann problem, modeling a molecule with three point charges. The surface of the molecule is such that the discontinuities do not align with the coarsest mesh or with

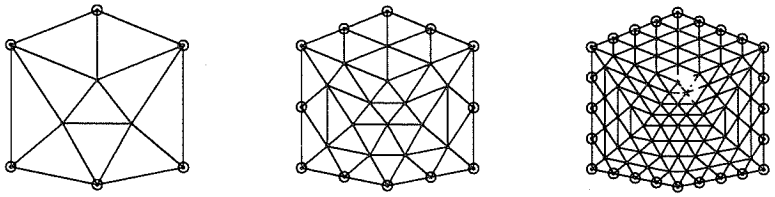


Figure 1: Example 1: Nested finite element meshes for MG.

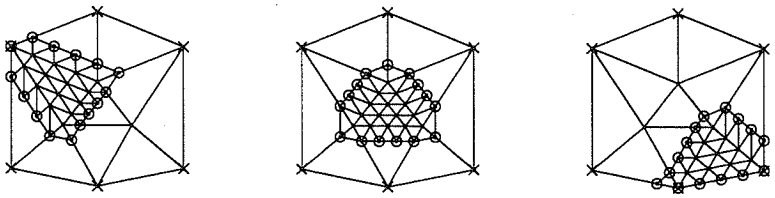


Figure 2: Example 1: Overlapping subdomains for DD.

the subdomain boundaries. Beginning with the coarse mesh shown on the left in Figure 1, we uniformly refine the initial mesh of 10 elements (9 nodes) five times. leading to a fine mesh of 2560 elements (1329 nodes). Piecewise linear finite elements, combined with one-point Gaussian quadrature, are used to discretize the problem. The three coarsest meshes used to formulate the MG methods are given in Figure 1. For the DD methods, the subdomains, corresponding to the initial coarse triangulation, are given a small overlap of one fine mesh triangle. The DD methods also employ a coarse space constructed from the initial triangulation. Figure 2 shows three overlapping subdomains overlaying the initial coarse mesh. Computed results are presented in Tables 1 to 4. Given for each experiment is the number of iterations required to satisfy the error criterion (reduction of the A-norm of the error by 10^{-10}). We report results for the unaccelerated, CG-accelerated, and Bi-CGstab-accelerated methods. The execution time differs for each method; normalized costs are tabulated in [5].

Multiplicative multigrid. The results for multiplicative V-cycle MG are presented in Table 1. Each row corresponds to a different smoothing strategy and is annotated by (ν_1, ν_2), with ν_1 pre-smoothing sweeps and ν_2 post-smoothing sweeps. An "f" indicates the use of a single forward Gauss-Seidel sweep, while a "b" denotes the use of the adjoint of the latter, i.e., a backward Gauss-Seidel sweep. Two series of results are given. For the first set, we explicitly imposed the Galerkin conditions when constructing the coarse operators. In this case, the multigrid algorithm is guaranteed to converge (cf. [5]). In the second series of tests (corresponding to the numbers in parentheses) the coarse mesh operators are constructed using standard finite element discretization. In that case, Galerkin conditions are not satisfied everywhere due to coefficient discontinuities appearing within coarse elements; hence, the MG method may diverge (DIV).

The unaccelerated MG results clearly illustrate the symmetry penalty given in Lemma 10.

Table 1: Example 1: Multiplicative MG with variational (discretized) coarse problem

ν_1	ν_2	UNACCEL	CG	Bi-CGstab
f	0	65 (DIV)	≫100 (≫100)	14 (16)
f	b	55 (DIV)	16 (18)	10 (15)
f	f	40 (31)	30 (≫100)	9 (9)
ff	0	39 (48)	≫100 (≫100)	8 (10)
fb	0	53 (DIV)	≫100 (≫100)	10 (11)
0	ff	39 (29)	29 (≫100)	8 (9)
0	fb	53 (DIV)	17 (99)	10 (12)
fb	fb	34 (27)	12 (13)	8 (8)
ff	bb	28 (18)	11 (11)	7 (7)
ff	ff	24 (15)	12 (12)	6 (6)
fff	f	24 (15)	17 (27)	6 (6)
ffff	0	25 (17)	≫100 (≫100)	7 (6)

Table 2: Example 1: Multiplicative DD with variational (discretized) coarse problem

Accel.	subdomain solve	forw	forw/back	forw/forw
UNACCEL	exact	40 (42)	38 (39)	20 (21)
	symmetric	279 (282)	146 (149)	140 (141)
	adjointed	–	110 (112)	102 (103)
	nonsymmetric	189 (191)	102 (104)	95 (96)
CG	exact	≫500 (≫500)	13 (13)	20 (20)
	symmetric	140 (56)	24 (24)	29 (27)
	adjointed	– –	21 (21)	25 (26)
	nonsymmetric	135 (83)	22 (23)	28 (28)
Bi-CGstab	exact	9 (9)	9 (9)	6 (6)
	symmetric	23 (23)	17 (16)	16 (16)
	adjointed	– –	14 (14)	14 (13)
	nonsymmetric	19 (20)	13 (13)	13 (13)

The nonsymmetric methods are always superior to the symmetric ones (the cases (f,b), (ff,bb), and (fb,fb)). Note that minimal symmetry (ff,bb) leads to a better convergence than maximal symmetry (fb,fb). The correctness of Lemma 10 is illustrated by noting that two iterations of the (f,0) strategy are actually faster than one iteration of the (f,b) strategy; also, compare the (ff,0) strategy to the (ff,bb) one. The CG-acceleration leads to a guaranteed reduction in iteration count for the symmetric preconditioners (see Lemma 12). We observe that the unaccelerated method need not be convergent for CG to be effective. CG appears to also accelerate some non-symmetric linear methods. Yet, it seems difficult to predict failure or success beforehand in such cases. The most robust method appears to be the Bi-CGstab method. Note the tendency

Table 3: Example 1: Additive MG with variational (discretized) coarse problem

ν	UNACCEL		CG		Bi-CGstab	
f	175	($\gg$1000)	$\gg$100	($\gg$100)	23	(52)
ff	110	($\gg$1000)	119	(168)	19	(43)
fb	146	($\gg$1000)	34	(54)	23	(49)
ffff	95	($\gg$1000)	28	(67)	17	(37)
ffbb	100	($\gg$1000)	27	(47)	17	(34)
fbfb	95	($\gg$1000)	28	(48)	20	(43)

Table 4: Example 1: Additive DD with variational (discretized) coarse problem

subdomain solve	UNACCEL		CG		Bi-CGstab	
exact	$\gg$1000	($\gg$1000)	34	(34)	25	(27)
symmetric	$\gg$1000	($\gg$1000)	57	(57)	50	(49)
nonsymmetric	$\gg$1000	($\gg$1000)	69	(65)	38	(41)

to favor the nonsymmetric V-cycle strategies. Overall, the fastest method proves to be the Bi-CGstab-acceleration of a (very nonsymmetric) V(1,0)-cycle.

Multiplicative domain decomposition. Results for multiplicative DD are given in Table 2. In the column "forw" the iteration counts reported were obtained with a single sweep though the subdomains on each multiplicative DD iteration. The other columns correspond to a symmetric forward/backward sweep or to two forward sweeps. Four different subdomain solvers are used: an *exact* solve, a *symmetric* method consisting of two symmetric Gauss-Seidel iterations, a *nonsymmetric* method consisting of four Gauss-Seidel iterations, and, finally, a method using four forward Gauss-Seidel iterations in the forward subdomain sweep and using their *adjoint* (i.e., four backward Gauss-Seidel iterations) in the backward subdomain sweep. The latter leads to a symmetric iteration; see Remark 2. Note that the cost of the three inexact subdomain solvers is identical.

Although apparently not as sensitive to operator symmetries as MG, the same conclusions can be drawn for DD as for MG. In particular, the symmetry penalty is seen for the pure DD results. Lemma 10 is confirmed since two iterations in the column "forw" are always more efficient than one iteration of the corresponding method in column "forw/back." The CG results indicate that using minimal symmetry (the "adjointed" column) is a more effective approach than the fully symmetric one (the "symmetric" column). The most robust acceleration is the Bi-CGstab one.

Additive multigrid. Results obtained with an additive multigrid method are reported in Table 3. The number and nature of the smoothing strategy is given in the first column of the table.

In the case of an unaccelerated additive method, the selection of a good damping param-

eter is crucial for convergence of the method. We did not search extensively for an optimal parameter; a selection of $\omega = 0.45$ seemed to provide good results in the case when the coarse problem was variationally defined. No ω-value leading to satisfactory convergence was found in the case when the coarse problems were obtained by discretization. In the case of CG acceleration the observed convergence behavior was completely independent of the choice of ω; see Remark 2. The symmetric methods ($\nu = fb, ffbb, fbfb$) are accelerated very well. Some of the nonsymmetric methods are accelerated too, especially when the number of smoothing steps is sufficiently large. The best method overall appears to be the Bi-CGstab acceleration of the nonsymmetric multigrid method with a single forward Gauss-Seidel sweep on each grid-level.

Additive domain decomposition. The results for additive DD are given in Table 4. The subdomain solver is either an exact solver, a symmetric solver based on two symmetric (forward/backward) Gauss-Seidel sweeps, or a nonsymmetric solver based on four forward Gauss-Seidel iterations. No value of ω was found that led to satisfactory convergence of the unaccelerated method. The CG-acceleration performs well when the linear method is symmetric and worse if nonsymmetric. Again, the best overall method is the Bi-CGstab-acceleration of the nonsymmetric additive solver.

REFERENCES

[1] S. F. Ashby, T. A. Manteuffel, and P. E. Saylor. A taxonomy for conjugate gradient methods. *SIAM J. Numer. Anal.*, 27(6):1542–1568, 1990.

[2] M. Dryja and O. B. Widlund. Towards a unified theory of domain decomposition algorithms for elliptic problems. In *Third International Symposium on Domain Decomposition Methods for Partial Differential Equations*, T. F. Chan, R. Glowinski, J. Périaux, and O. B. Widlund, eds. SIAM, Philadelphia, PA, pp. 3-21, 1989.

[3] W. Hackbusch. *Iterative Solution of Large Sparse Systems of Equations.* Springer-Verlag, Berlin, Germany, 1994.

[4] M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *J. Research of NBS*, 49:409–435, 1952.

[5] M. Holst and S. Vandewalle. Schwarz methods: to symmetrize or not to symmetrize. *SIAM J. Numer. Anal.* (to appear).

[6] Y. Saad and M. H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Statist. Comput.*, 7(3):856–869, 1986.

[7] P. Sonneveld. CGS: A fast Lanczos-type solver for nonsymmetric linear systems. *SIAM J. Sci. Statist. Comput.*, 10:36–52, 1989.

[8] H. A. van der Vorst. BI-CGSTAB: A fast and smoothly converging variant of BI-CG for the solution of nonsymmetric linear systems. *SIAM J. Sci. Statist. Comput.*, 13(2):631–644, 1992.

[9] J. Xu. *Theory of Multilevel Methods.* Ph.D. thesis, Technical Report AM 48, Department of Mathematics, Penn State University, University Park, PA, July 1989.

[10] J. Xu. Iterative methods by space decomposition and subspace correction. *SIAM Review*, 34(4):581–613, 1992.

A Mixed Finite Volume Element Method for Flow Calculations in Porous Media

Jim E. Jones
Institute for Computer Applications in Science and Engineering
NASA Langley Research Center

SUMMARY

A key ingredient in the simulation of flow in porous media is the accurate determination of the velocities that drive the flow. The large scale irregularities of the geology, such as faults, fractures, and layers suggest the use of irregular grids in the simulation. Work has been done in applying the finite volume element (FVE) methodology as developed by McCormick in conjunction with mixed methods which were developed by Raviart and Thomas. The resulting mixed finite volume element discretization scheme has the potential to generate more accurate solutions than standard approaches. The focus of this paper is on a multilevel algorithm for solving the discrete mixed FVE equations. The algorithm uses a standard cell centered finite difference scheme as the 'coarse' level and the more accurate mixed FVE scheme as the 'fine' level. The algorithm appears to have potential as a fast solver for large size simulations of flow in porous media.

The Mixed Finite Volume Element Discretization

In this first section, we briefly introduce the mixed finite volume element (FVE) discretization technique. We will not dwell too much on the details of the discretization itself as our focus here is on solving the discrete set of equations that the discretization produces; a detailed description of the discretization can be found in [7].

We begin by considering the following partial differential equation defined on a domain Ω in $\mathcal{R}^2$:

$$\begin{cases} -\nabla \cdot \mathbf{A}(\mathbf{x})\nabla\phi(\mathbf{x}) = f(\mathbf{x}) & \mathbf{x} \in \Omega, \\ \nabla\phi(\mathbf{x}) \cdot \eta = g(\mathbf{x}) & \mathbf{x} \in \partial\Omega. \end{cases} \tag{1}$$

Here we assume the diffusion coefficient $\mathbf{A}$ is diagonal, but values of the coefficients may jump orders of magnitude at material interfaces. In the context of reservoir simulation, this is the pressure equation for incompressible single-phase flow where ϕ is the pressure in the reservoir Ω, and the boundary condition specifies the flux

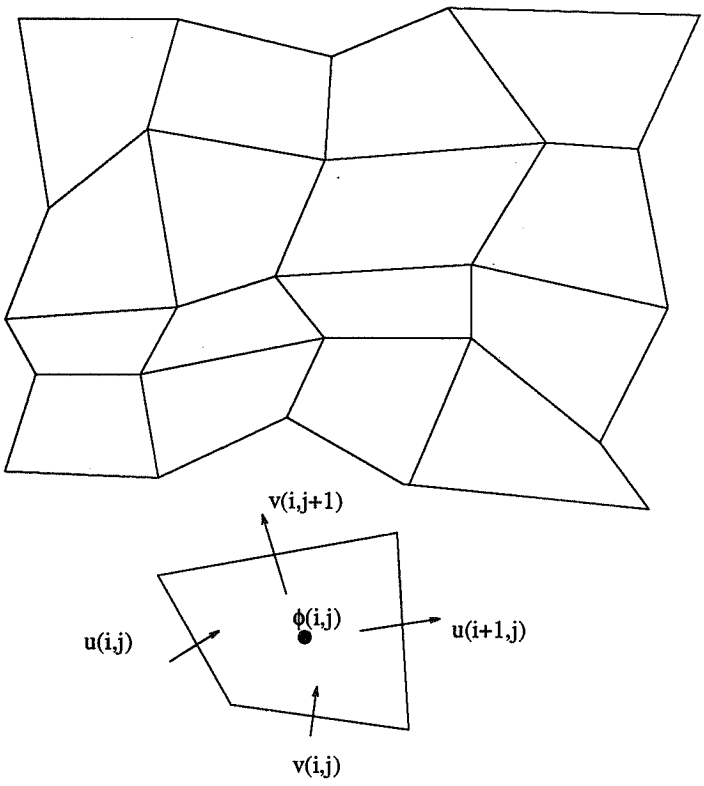


Figure 1

on $\partial\Omega$. As one of our goals for the new discretization is accurate approximations of flow velocities, we will begin by reformulating this equation as a first order system of equations where velocity appears explicitly in the equations. This is done by introducing the flow velocity variables via the definition,

$$\mathbf{v} \equiv -\mathbf{A}\nabla\phi, \tag{2}$$

and then rewriting the partial differential equation in (1) as,

$$\nabla \cdot \mathbf{v} = f. \tag{3}$$

In the context of reservoir simulation, definition (2) is Darcy's law and equation (3) is the mass conservation law. In reservoir simulation, this same approach of treating flow velocity explicitly has been used in mixed finite-element methods with considerable success [5],[6],[13]. Equations (2) and (3) along with the boundary condition from equation (1) represent the first order system that we discretize using the mixed FVE method. Because of the irregularity of reservoir geology, faults, layers, etc., uniform rectangular grids are not adequate in modeling the flow. The mixed FVE discretization was developed for a logically rectangular grid of irregular quadrilaterals. An example of such a grid is shown in Figure 1. To discretize this system, we follow the finite volume element (FVE) principles developed in [3],[8],[9]. The two major components of any FVE discretization scheme are a choice of control volumes

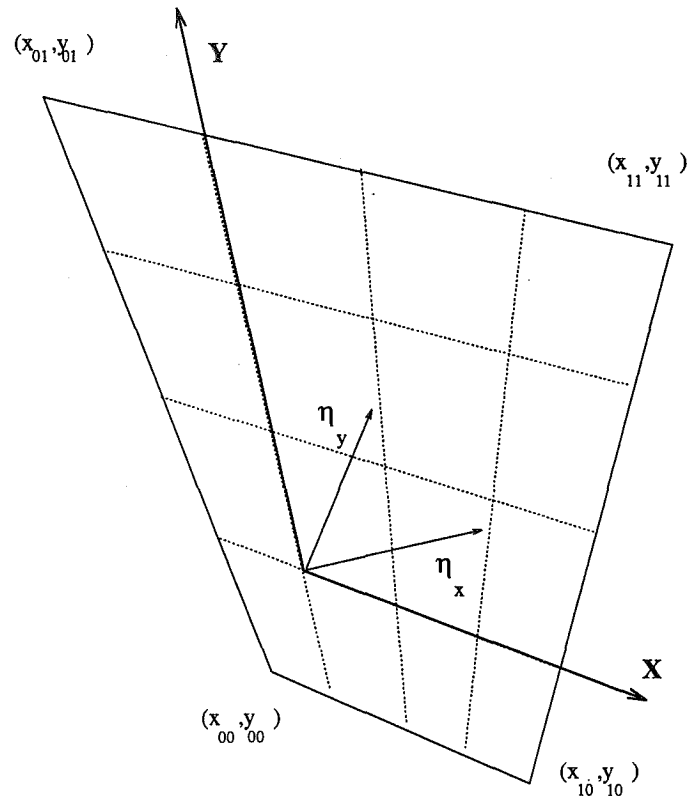


Figure 2

to integrate the continuous equation over and a choice of finite element spaces for the unknowns.

Important in developing the discretization for general quadrilaterals is the mapping relating a general quadrilateral to a reference one. Consider the quadrilateral P with vertices $(x_{00}, y_{00}), (x_{10}, y_{10}), (x_{01}, y_{01})$, and (x_{11}, y_{11}) shown in Figure 2. Let the reference quadrilateral $\hat{P}$ be the unit square. Then there is a unique bilinear mapping of $\hat{P}$ onto P given by,

$$x(\hat{x}, \hat{y}) = x_{00} + (x_{10} - x_{00})\hat{x} + (x_{01} - x_{00})\hat{y} + (x_{11} - x_{10} - x_{01} + x_{00})\hat{x}\hat{y}$$
$$y(\hat{x}, \hat{y}) = y_{00} + (y_{10} - y_{00})\hat{x} + (y_{01} - y_{00})\hat{y} + (y_{11} - y_{10} - y_{01} + y_{00})\hat{x}\hat{y}$$

If P is convex, then this mapping has an inverse. We restrict ourselves to convex quadrilaterals, so for each $(x, y) \in P$ we have an associated point $(\hat{x}, \hat{y}) \in \hat{P}$. Shown in Figure 2 are several vectors that will be useful later in describing the components of our discretization technique. For each $(x, y) \in P$ we define four vectors.

$\mathbf{X}(x, y)$ is the image of the unit vector $(1, 0)$ in $\hat{P}$,
$\mathbf{Y}(x, y)$ is the image of the unit vector $(0, 1)$ in $\hat{P}$,
$\eta_x(x, y)$ is a unit vector orthogonal to $\mathbf{Y}(x, y)$,
$\eta_y(x, y)$ is a unit vector orthogonal to $\mathbf{X}(x, y)$.

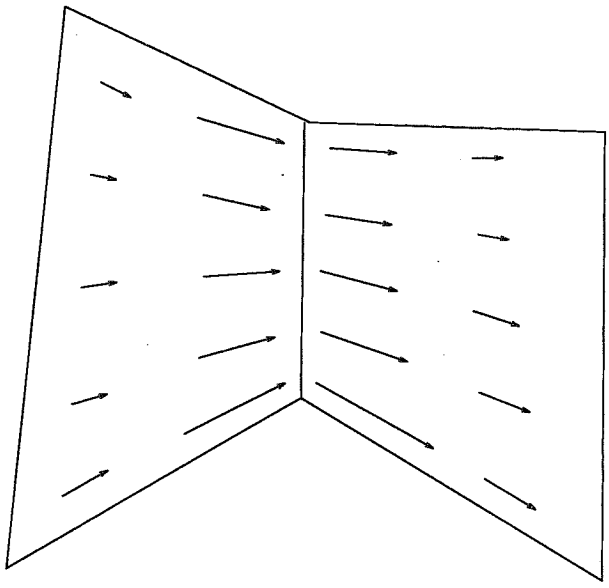


Figure 3

For the finite element spaces we use the lowest order Raviart-Thomas elements on the quadrilateral elements, see [2],[14] and [11]. They can be defined as follows. The characteristic functions of the quadrilaterals provide a basis for the finite element space for ϕ. The basis functions for $\mathbf{v}$ are best seen by associating degrees of freedom with normal components on edges of quadrilaterals. A typical basis function for the finite element space for $\mathbf{v}$ has support on two adjacent quadrilaterals and has a constant normal component on the edge shared by the quadrilaterals, and its normal component is zero on other edges. The magnitude of the basis function is such that the flux on the common edge is one,

$$\int_{edge} \mathbf{v} \cdot \eta ds = 1.$$

These conditions alone do not uniquely determine the basis function; the following additional condition on the finite element space is needed. Within any quadrilateral P,

$$\begin{array}{ll} \mathbf{v} \cdot \eta_x \|\mathbf{Y}\| & \text{varies linearly with } \hat{x}, \text{ constant with } \hat{y}, \\ \mathbf{v} \cdot \eta_y \|\mathbf{X}\| & \text{varies linearly with } \hat{y}, \text{ constant with } \hat{x}. \end{array}$$

A typical basis function is represented in Figure 3. We note that the basis functions have continuous normal components across grid interfaces. With this we can guarantee that our computed flow velocity will also have continuous normal component across grid edges. The true physical solution also has this property, continuous normal component of velocities, but not every numerical scheme for approximating it does, as pointed out in [12].

We now need to choose the control volumes. The quadrilaterals used to describe the grid are the natural choice for the control volumes for equation (3). This will

produce a scheme with a local conservation property on these quadrilateral grid cells. So we integrate equation (3), over each grid cell $\mathbf{P}_{i,j}$,

$$\int_{\mathbf{P}_{i,j}} \nabla \cdot \mathbf{v} dx dy = \int_{\mathbf{P}_{i,j}} f.$$

Applying the divergence theorem, we get,

$$\int_{\partial \mathbf{P}_{i,j}} \mathbf{v} \cdot \eta ds = \int_{\mathbf{P}_{i,j}} f.$$

The left-hand side of this equation is just the sum of the fluxes on edges of $\mathbf{P}_{i,j}$, so the discretization of the mass conservation equation is,

$$u^h_{i+1,j} - u^h_{i,j} + v^h_{i,j+1} - v^h_{i,j} = \int_{\mathbf{P}_{i,j}} f. \tag{4}$$

Here $u^h_{i+1,j}$ and $u^h_{i,j}$ denote the discrete fluxes on the 'east' and 'west' edges of the grid cell, respectively. Similarly, $v^h_{i,j+1}$ and $v^h_{i,j}$ denote the discrete fluxes on the 'north' and 'south' edges of the grid cell, respectively. If we assume f is (approximated by) a function that is piecewise constant, we can replace the integral on the right hand side by:

$$f_{i,j} \times AREA\,(\mathbf{P}_{i,j})\,.$$

If we have more information about f, we can use a more accurate approximation of the integral. In choosing the control volumes for Darcy's equation, we use the following control volumes which straddle grid edges. Consider two adjacent grid cells, $\mathbf{P}_{i-1,j}$ and $\mathbf{P}_{i,j}$. $\mathbf{U}_{i,j}$ then consists of the image of $(1/2,1) \times (0,1)$ under the mapping for $\mathbf{P}_{i-1,j}$ and the image of $(0,1/2) \times (0,1)$ under the mapping for $\mathbf{P}_{i,j}$. In Figure 4, $\mathbf{U}_{i,j}$ is the shaded region. We associate this volume with the 'vertical' edge shared by $\mathbf{P}_{i-1,j}$ and $\mathbf{P}_{i,j}$ which the control volume straddles. We also have control volumes associated with 'horizontal' edges. For adjacent grid cells, $\mathbf{P}_{i,j-1}$ and $\mathbf{P}_{i,j}$, $\mathbf{V}_{i,j}$ consists of the image of $(0,1) \times (1/2,1)$ under the mapping for $\mathbf{P}_{i,j-1}$ and the image of $(0,1) \times (0,1/2)$ under the mapping for $\mathbf{P}_{i,j}$. The discretization of Darcy's equation proceeds as follows. We dot equation (2) with $c_l\mathbf{X}(x,y)$ and integrate over the 'left half' of $\mathbf{U}_{i,j}$. Similarly, we dot equation (2) with $c_r\mathbf{X}(x,y)$ and integrate over the 'right half' of $\mathbf{U}_{i,j}$. Here c_l and c_r are scaling constants chosen in such a way to eliminate integral terms on the interface between $\mathbf{P}_{i-1,j}$ and $\mathbf{P}_{i,j}$ where ϕ^h is undefined. We then add the two integrals to get our final result. We will present here only the form of the equation that this integration gives rise to. Note that we perform the same kind of integrations for the V volumes as well, only here we dot Darcy's law with a scaling of the vector $\mathbf{Y}$. For the U volume shown in Figure 4, we get a discrete Darcy equation relating the pressure drop between the two cells to the fluxes on cell edges,

$$\begin{aligned} &c_1 u_{i-1,j} + c_2 u_{i,j} \\ &+c_3 u_{i+1,j} + c_4 v_{i-1,j} + c_5 v_{i-1,j+1} + c_6 v_{i,j} + c_7 v_{i,j+1} \\ &+|E|(\phi_{i,j} - \phi_{i-1,j}) = 0. \end{aligned} \tag{5}$$

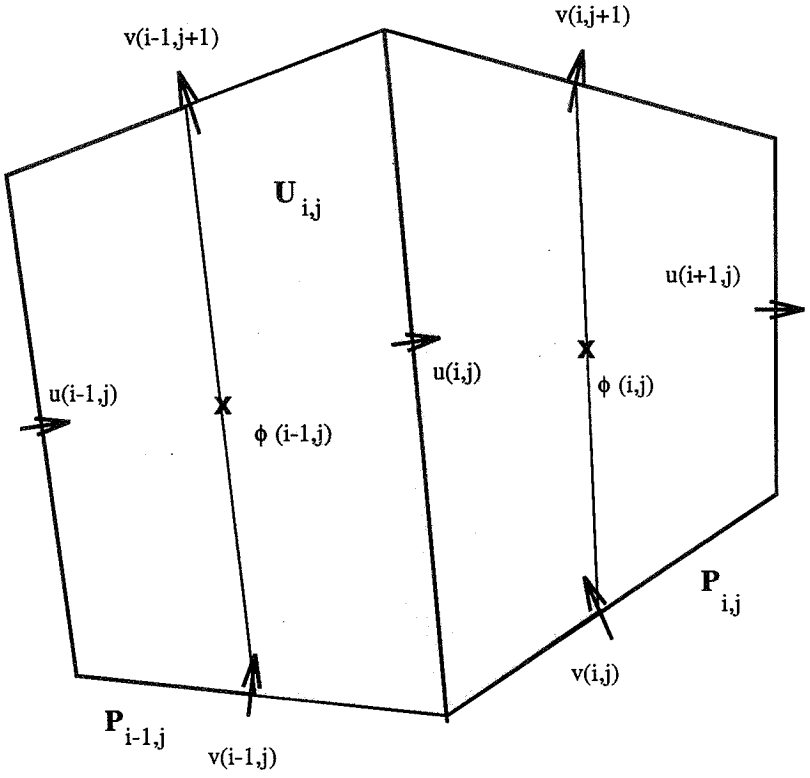


Figure 4

Here $|E|$ is the length of the edge shared by the two adjacent grid cells. The values of the coefficients $c_1, \ldots, c_7$ depend on the position of the vertices defining the two grid cells and on the values of the diffusion coefficient within the two cells. The 'cross' terms, $c_4, \ldots, c_7$, will generally be nonzero even when the diffusion coefficient $\mathbf{A}$ is diagonal. In summary, for each grid cell we have a discrete conservation equation of the form of equation (4) and for each grid edge we have a discrete Darcy equation of the form of equation (5).

A Multilevel Algorithm

Previously in [7], a multigrid algorithm was developed to solve the discrete set of equations that the mixed FVE method produces. In this algorithm the mixed FVE discretization was used on coarser levels and interpolation and restriction were done in a way consistent with the finite element spaces and control volumes on different grids as in [9]. This yields a very efficient algorithm with two limitations. The first is that the jumps in the diffusion coefficient must occur (if at all) at grid edges on the coarse grid. The second is that the irregularity is described in a coarse grid which is then refined by bilinear coordinates to generate finer grids. One cannot apply this mixed FVE based multigrid algorithm to the equations on the coarsest grid; they must be solved some other way. In a practice, both these problems are limitations

on the coarsest grid; it must be fine enough to capture the reservoir geology and the jumps in the diffusion coefficient. These limitations may result in the set of discrete equations on the coarsest grid being too large to solve directly. With these limitations we were forced to seek an alternative algorithm, either to be used alone or as the solver on the coarsest grid allowable in a mixed FVE based multigrid algorithm.

We will explain our approach as a two level multigrid algorithm; this is somewhat an incorrect name as we will have only one grid. The fine level problem is the mixed FVE discretization of the first order system,

$$\begin{aligned} \mathbf{A}^{-1}\mathbf{v} + \nabla\phi &= 0, \\ \nabla \cdot \mathbf{v} &= f. \end{aligned}$$

We will write the mixed FVE equations in matrix form as,

$$\begin{pmatrix} M & grad_h \\ div_h & 0 \end{pmatrix} \begin{pmatrix} \mathbf{v}_h \\ \phi_h \end{pmatrix} = \begin{pmatrix} 0 \\ f_h \end{pmatrix}. \tag{6}$$

Here, M is the mass matrix that comes from the discretization of Darcy's equation and $grad_h$ and div_h are the grid h discrete operators corresponding to the continuous operators $grad$ and div. We define the residuals as,

$$\begin{pmatrix} r^h_{\mathbf{v}} \\ r^h_{\phi} \end{pmatrix} = \begin{pmatrix} 0 \\ f_h \end{pmatrix} - \begin{pmatrix} M & grad_h \\ div_h & 0 \end{pmatrix} \begin{pmatrix} \hat{\mathbf{v}}_h \\ \hat{\phi}_h \end{pmatrix} \tag{7}$$

where the variables with hats denote a current approximate solution to equation (6). We define the errors as,

$$\begin{aligned} e^h_{\mathbf{v}} &= \mathbf{v}^h - \hat{\mathbf{v}}^h, \\ e^h_{\phi} &= \phi^h - \hat{\phi}^h. \end{aligned}$$

We then write the error equation,

$$\begin{pmatrix} M & grad_h \\ div_h & 0 \end{pmatrix} \begin{pmatrix} e^h_{\mathbf{v}} \\ e^h_{\phi} \end{pmatrix} = \begin{pmatrix} r^h_{\mathbf{v}} \\ r^h_{\phi} \end{pmatrix}. \tag{8}$$

Now rather than using a coarser grid with the mixed FVE discretization to approximate the error equation, we will use the same grid with a standard cell-centered finite difference approximation. This will be our 'coarse' level in the multigrid algorithm. The 'coarse' level version of the error equation can then be written in matrix form as,

$$\begin{pmatrix} \tilde{M} & grad_h \\ div_h & 0 \end{pmatrix} \begin{pmatrix} \mathbf{v}^h \\ \phi^h \end{pmatrix} = \begin{pmatrix} r^h_{\mathbf{v}} \\ r^h_{\phi} \end{pmatrix}. \tag{9}$$

The only difference between equations (8) and (9) is the mass matrix. Assume the grid is rectangular and the diffusion coefficient is diagonal,

$$\mathbf{A} = \begin{pmatrix} a_x & 0 \\ 0 & a_y \end{pmatrix}.$$

Then in (9) the mass matrix $\tilde{M}$ is diagonal and is computed from the diffusion coefficient by

$$u^h_{i,j} = \frac{a_x^{-1}}{h}\left(\phi^h_{i,j} - \phi^h_{i-1,j}\right) = 0, \tag{10}$$

The point is that in equation (9) we can eliminate the velocity variables. We have,

$$\mathbf{v}^h = \tilde{M}^{-1}\left(grad_h\phi_h + r^h_{\mathbf{v}}\right). \tag{11}$$

Using this we can write equation (9) as,

$$-div_h\tilde{M}^{-1}grad_h\phi_h = r^h_\phi - div_h\tilde{M}^{-1}r^h_{\mathbf{v}}. \tag{12}$$

Black box multigrid [4] was developed to solve precisely this type of equation. In the multilevel solver, we use black box multigrid to solve this equation for ϕ, use equation (11) to get $\mathbf{v}^h$, and use these approximations of the error to correct our mixed FVE approximation. In summary when the grid is rectangular, we can use a standard cell centered finite difference discretization as the 'coarse' level for the 'fine' level mixed FVE discretization. We would like do something similar in the case of a general quadrilateral grid. However, one of our motivations for looking at the mixed FVE discretization was that it can be applied in a clear and direct way to general quadrilateral grids where standard cell centered finite differences cannot. A rigorous cell centered finite difference discretization for general quadrilateral grids does not currently exist. Fortunately, we do not need to ask this much of the discretization on the 'coarse' level as we will use the solution from the mixed FVE discretization for our final computation. We would like to use the 'coarse' level discretization only to accelerate the relaxation process on the 'fine' level. We have chosen to use equation (10) to define $\tilde{M}$ in the general quadrilateral case just as in the uniform case. There are perhaps more sophisticated ways of defining $\tilde{M}$, but we have found that this simple definition works well for most grids. It is clear that for very distorted grids, our $\tilde{M}$ will be a poor approximation to M; however, we will see in the next section that for mildly distorted grids the two level method works as well as in the uniform grid case. This two level approach is similar to the work in [10] where black box multigrid was used as a 'coarse' level for a Lagrangian hydrodynamics application.

Computational Results

Problem 1

We begin with a test problem using a uniform square grid on $\Omega = [-1,1]\times[-1,1]$. The numerical experiment is designed to test the robustness of the two level approach with respect to discontinuities in the diffusion coefficient. The diffusion coefficients,

a_x and a_y, were separately and randomly assigned values between .01 and 100 for each grid cell. The problem is thus anisotropic and the coefficients jump several orders of magnitude between cells. The two level algorithm described in the previous section was used with two alternating line relaxation sweeps on the mixed FVE equations before calling in black box multigrid to solve the 'coarse' problem and one alternating line relaxation sweep after. Here x-line relaxation, for instance, means changing all variables, ϕ, u, and v, associated with cells sharing the same j index so that all discrete equations (conservation and Darcy) associated with those cells are satisfied. This involves inverting a banded $4n-3$ by $4n-3$ matrix, where n is the number of cells in the x-direction. This is a relatively expensive relaxation process, but it is needed to deal with anisotropic coefficients. As pointed out in [1], block relaxation is needed for smoothing when cells are coupled to some neighboring cells strongly and to other neighboring cells weakly. One point to consider is: how well do we need black box multigrid to solve the 'coarse' problem? In [10] only one cycle of black box multigrid was used to approximately solve the 'coarse' problem. We found that for this difficult test problem (note: black box multigrid convergence factors were approximately equal to .6) that performing more than one cycle of black box multigrid improved the overall convergence factors of the two level method. In the results reported below we used five cycles of black box multigrid to approximately solve the 'coarse' problem, although similar multilevel convergence factors can be obtained with fewer (say, two or three) cycles. The asymptotic convergence factors for the two level method are presented below.

Grid size	Convergence Factor
16×16	.43
32×32	.44
64×64	.46

We see that this two level approach, while not having great convergence factors, does exhibit convergence factors that are constant with growing problem size. The point of considering this two level method was to allow us to deal with the problem where the coarsest grid for the mixed FVE based multigrid algorithm is still too fine and has too many unknowns to solve the mixed FVE equations using a direct method. In practice, one could use the mixed FVE based multigrid algorithm until one reached the coarsest grid that was aligned with the discontinuities in the diffusion coefficient. Then, on this grid, use the two level approach of the previous section.

Problem 2

In this experiment we began with a uniform square grid on $\Omega = [-1,1] \times [-1,1]$ and distorted the grid in the following way. We moved each interior vertex in both the x and y directions separately by a random number between $-.2h$ and $.2h$, where h was the mesh size of the original square mesh. The resulting mesh for the 16×16 problem

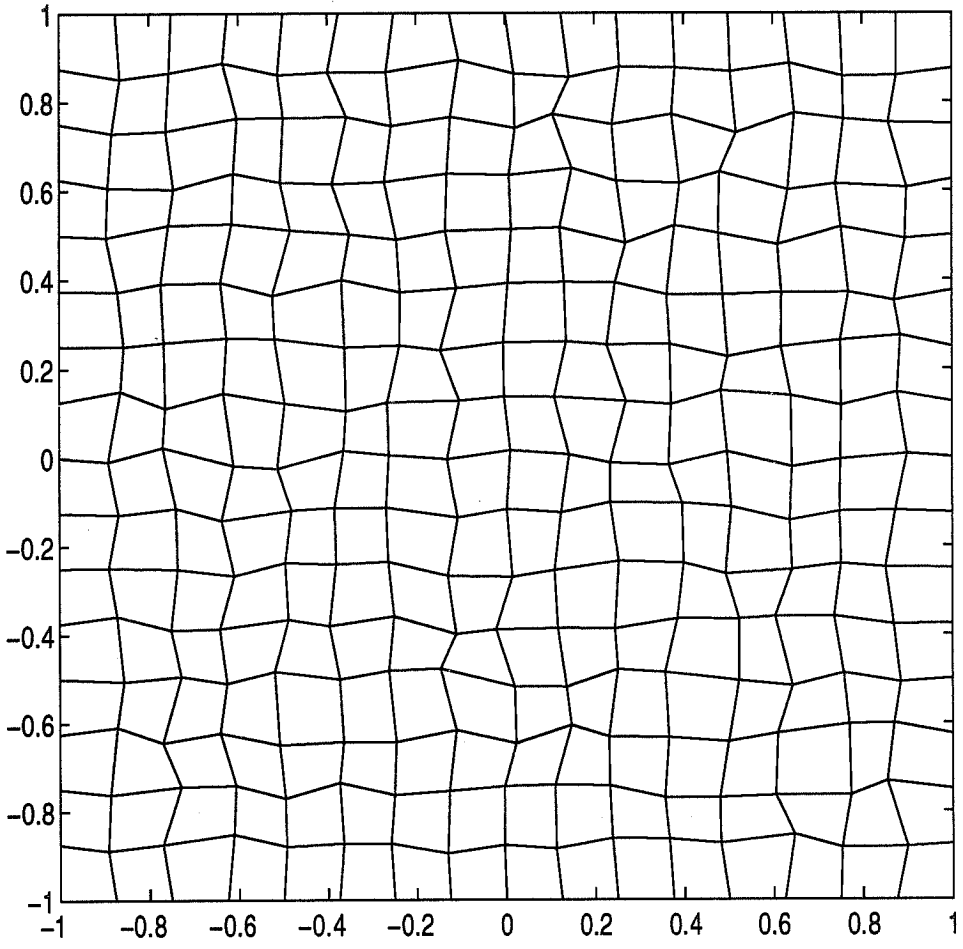


Figure 5

is shown in Figure 5. Then we discretized Poisson's equation, $a_x = a_y = 1$, using the mixed FVE method and applied the two level algorithm described previously. Again, two alternating line relaxation sweeps were performed before solving the 'coarse' problem and one alternating line relaxation sweep was performed after, and five cycles of black box multigrid were used to solve the 'coarse' problem. Average convergence factors for the two level approach on different size grids are shown below.

Grid size	Convergence Factor
16×16	.07
32×32	.08
64×64	.08

The convergence factors are surprisingly good, given the quality of the approximation used on the 'coarse' level. As discussed previously, we basically assume the grid is uniform in forming the mass matrix $\tilde{M}$ for the discrete Darcy equations on the 'coarse' level. This appears to work fine for the mildly distorted grids like the grids in this numerical experiment and, quite likely, the grids one would use in practical applications. When the grid is very distorted, say 50% rather than 20% distortion, the two level algorithm can fail to converge and may even diverge. The reason is that the very poor approximation of the mass matrix results in a correction from the 'coarse' level that has little, if anything, to do with the 'fine' level error. It is possible that this could be remedied by a more sophisticated choice for $\tilde{M}$, but this has not been investigated.

Problem 3

In the next numerical experiment we use the same grids as in the previous experiment and solve the mixed FVE discretization to the diffusion equation with diagonal diffusion coefficient where on each cell in this grid the diffusion coefficients a_x and a_y were separately set to random values between .01 and 100. The results, average convergence factors, are shown below.

Grid size	Convergence Factor
16×16	.43
32×32	.40
64×64	.38

While the convergence factors are not that great, they likely are acceptable especially if one is using the two level approach only on the coarsest grid of the mixed FVE based multigrid algorithm. There the amount of work on finer grids in the mixed FVE based multigrid algorithm will be much larger than the work of the two level algorithm on the coarsest grid, even if several cycles of the two level algorithm are required. This last experiment is reflective of the types of problems one would solve in actual reservoir simulation. It appears that this two level approach has the

potential to provide a fast solution to the more accurate mixed FVE discretization, compared to standard cell centered finite differences, in cases where the previously developed mixed FVE based multigrid algorithm cannot be applied.

Conclusions

The two level algorithm presented in this paper provides an efficient method for solving the mixed FVE equations on general quadrilateral grids. One point about the "poor" convergence factors for the two level method seen in problems 1 and 3: these results, in an indirect way, illustrate the superiority of the mixed FVE discretization over the standard cell centered finite difference discretization when the diffusion coefficient is discontinuous, even on uniform grids. The "poor" convergence factors tell us that there is a significant difference between the discretizations, and as demonstrated in [7], the mixed FVE discretization is the more accurate of the two.

REFERENCES

[1] A. Brandt, *Multigrid Techniques : 1984 Guide*, The Weizmann Institute of Science, Rehovot, Israel, 1984.

[2] F. Brezzi and M. Fortin, *Mixed and Hybrid Finite Element Methods*, Springer Series in Computational Mathematics Number 15, Springer-Verlag, 1991.

[3] Z. Cai, J. Mandel, and S.F. McCormick, "The finite volume element method for diffusion equations on general triangulations," *SIAM Journal of Numerical Analysis, 28*, 1991, 392-402.

[4] J.E. Dendy, "Black box multigrid," *Journal of Computational Physics, 48*, 1982, 366-386.

[5] R.E. Ewing and R.F. Heinemann, "Incorporation of mixed finite element methods in compositional simulation for reduction of numerical dispersion," SPE 12267, *Proceedings of the 7th SPE Symposium on Reservoir Simulation*, 1983, pp. 341-347.

[6] R.E. Ewing, T.F. Russell, and M.F. Wheeler, "Simulation of miscible displacement using mixed methods and a modified method of characteristics," SPE 12241, *Proceedings of the 7th SPE Symposium on Reservoir Simulation*, 1983, pp. 71-81.

[7] J. Jones, "A mixed finite volume element method for accurate computation of fluid velocities in porous media," Ph.D Thesis, University of Colorado at Denver, 1995.

[8] C. Liu and S.F. McCormick, "The finite volume element method (FVE) for planar cavity flow," *Proceedings of the 11th International Conference on CFD*, Williamsburg, VA, June 28-July 2, 1988.

[9] S.F. McCormick, *Multilevel Adaptive Methods for Partial Differential Equations*, Vol. 6 in Frontiers in Applied Mathematics, Society for Industrial and Applied Mathematics, Philadelphia, 1989.

[10] J.E. Morel, J.E. Dendy, M.L. Hall, and S.W. White, "A cell-centered Lagrangian-mesh diffusion differencing scheme," *Journal of Computational Physics, 103*, 1992, pp. 286-299.

[11] P.A. Raviart and J.M. Thomas, "A mixed finite element method for 2nd order elliptic problems," *Mathematical Aspects of Finite Element Methods*, I. Galligani and E. Magenes, eds., Lecture Notes in Mathematics, Vol. 606, Springer-Verlag, 1977, pp. 292-315.

[12] T.F. Russell, "Rigorous block-centered discretizations on irregular grids," *Improved Simulation of Complex Reservoir Systems*, Reservoir Simulation Research Corporation, Project Report No. 2, April 12, 1990.

[13] T.F. Russell and M.F. Wheeler, "Finite element and finite difference methods for continuous flows in porous media," *The Mathematics of Reservoir Simulation*, R.E. Ewing, ed., Society for Industrial and Applied Mathematics, Philadelphia, 1983, pp. 35-106.

[14] J.M. Thomas, "Sur l'analyse numerique des methodes d'elements finis hybrides et mixtes," These de doctorat d'etat, à l'Université Pierre et Marie Curie, May 1977.

Page intentionally left blank

Implicit Extrapolation Methods for Variable Coefficient Problems

M. Jung

U. Rüde

SUMMARY

Implicit extrapolation methods for the solution of partial differential equations are based on applying the extrapolation principle *indirectly*. Multigrid tau-extrapolation is a special case of this idea. In the context of multilevel finite element methods, an algorithm of this type can be used to raise the approximation order, even when the meshes are nonuniform or locally refined. Here previous results are generalized to the variable coefficient case and thus become applicable for nonlinear problems. The implicit extrapolation multigrid algorithm converges to the solution of a higher order finite element system. This is obtained without explicitly constructing higher order stiffness matrices but by applying extrapolation in a natural form within the algorithm. The algorithm requires only a small change of a basic low order multigrid method.

Introduction

Implicit extrapolation is an efficient technique to improve the accuracy of a multilevel solver. When combined with extrapolation, the multilevel principle is not only used as the basis for a fast algebraic solver, but also to increase the approximation order. The basic idea of extrapolation is to exploit discretizations on different levels.

In classical Richardson extrapolation, two or more approximations from different meshes are combined linearly to eliminate the dominating terms of the error expansion. For partial differential equations this has been studied in the context of finite difference discretizations, see e.g. Marchuk and Shaidurov [1] and in the framework of finite elements (FE), see e.g. Blum, Lin, and Rannacher [2]. These techniques are *explicit* extrapolation methods, since they use approximate solutions *directly*.

Here we propose a different approach, where extrapolation is applied *indirectly* to intermediate quantities of the solution process. Such methods are called *implicit* extrapolation techniques. Methods of this type may be related to defect correction, and — if combined with multigrid — to τ-extrapolation, see e.g. Brandt [3], Hackbusch [4], Schaffer [5], or Bernert [6]. However, these methods are mathematically still motivated by expansions of the truncation error, which in turn require uniform

meshes. A generalization to locally uniform meshes can e.g. be found in McCormick and Rüde [7].

In Jung and Rüde [8] we have presented an implicit finite element extrapolation technique which is based on extrapolating the quadrature rules used to compute the stiffness matrices. In [8] it has been shown that within the nested spaces of a multilevel finite element algorithm, this implicit extrapolation converts an h–hierarchical to a p–hierarchical basis. This improves the approximation order, independent of any uniformity constraints on the mesh and without requiring global asymptotic error expansions. On the other hand, the algorithm presented in [8] is algebraically just a special case of multigrid τ-extrapolation, which differs from the usual multilevel process only by an additional factor appearing in the restriction of the residual. The method is therefore particularly convenient to implement in any given multigrid algorithm.

The analysis of [8] was still restricted to problems with element-wise constant coefficients. In this present paper we will now generalize these results to show that an analogous algorithm can be used for variable coefficients as long as the coefficients are smooth enough to justify higher order approximations at all. The analysis is again based on studying quadrature formulas for the stiffness matrices, and using extrapolation to construct quadrature formulas which are exact for higher order polynomial functions. For variable coefficients, this is now significantly more complicated and our analysis requires nonstandard quadrature rules. These rules and the multilevel algorithm are introduced in detail. The final section presents a numerical example showing the efficiency of the method.

The boundary value problem and its finite element discretization

In this paper we consider two–dimensional second order elliptic boundary value problems given in the weak formulation

$$\text{Find } u \in V_0 \text{ such that } \quad a(u,v) = \langle F, v\rangle \quad \text{for all } v \in V_0, \tag{1}$$

with

$$a(u,v) = \int_\Omega (A(x)\nabla_x u, \nabla_x v)\, dx \tag{2}$$

and

$$\langle F, v\rangle = \int_\Omega f v\, dx. \tag{3}$$

Ω is a two-dimensional bounded polygonal domain. The space $V_0 = H_0^1(\Omega)$ is a subspace of the Sobolev space $H^1(\Omega)$, where the functions of V_0 satisfy homogeneous Dirichlet boundary conditions on the boundary $\partial\Omega$. The restriction to this type of boundary conditions is only to keep the exposition as simple as possible. The generalization to somewhat more general boundary conditions is analogous to [8].

Furthermore, we suppose that the 2×2 matrix $A(x) = (a_{ij}(x))_{i,j=1,2}$ is symmetric and positive definite for almost all $x \in \Omega$ with $a_{ij}(x) \in W_\infty^2(\Omega)$. The function f

belongs to the space $W_q^2(\Omega)$ with $q \geq 2$. We need these assumptions to obtain a discretization error which is typical for FE discretizations with piecewise quadratic functions and the application of appropriate quadrature rules for the computation of the stiffness matrix and the load vector.

We now discretize (1) by three different finite element spaces. We suppose that two nested triangulations $\mathcal{T}_{l-1}$ and $\mathcal{T}_l$ of the domain Ω are given. The finer triangulation $\mathcal{T}_l$ results from $\mathcal{T}_{l-1}$ by regular refinement, that is by connecting the midpoints of all triangles $\delta_{l-1}^{(r)}$, $r = 1, 2, \ldots, R_{l-1}$, in $\mathcal{T}_{l-1}$. Corresponding to the triangulations $\mathcal{T}_{l-1}$ and $\mathcal{T}_l$ we introduce the finite element spaces

$$V_{l-1}^L = \text{span}\{p_{l-1}^{(i)} : i = 1, 2, \ldots, N_{l-1}\} \subset V_0, \tag{4}$$

$$V_l^L = V_{l-1}^L \cup \text{span}\{p_l^{(i)} : i = N_{l-1}+1, \ldots, N_l\} \subset V_0, \tag{5}$$

$$V_l^Q = V_{l-1}^L \cup \text{span}\{q_{l-1}^{(i)} : i = N_{l-1}+1, \ldots, N_l\} \subset V_0. \tag{6}$$

The trial functions $p_k^{(i)}$, $k = l, l-1$, are continuous and piecewise linear in each triangle of $\mathcal{T}_k$ and they satisfy

$$\begin{aligned} p_{l-1}^{(i)}(x^{(j)}) &= \delta_{ij} \quad \text{for } i,j = 1, 2, \ldots, N_{l-1} \\ p_l^{(i)}(x^{(j)}) &= \delta_{ij} \quad \text{for } i,j = N_{l-1}+1, \ldots, N_l. \end{aligned}$$

Here $x^{(j)} = (x_1^{(j)}, x_2^{(j)})$ denotes the coordinates of the node $P^{(j)}$ and N_k is the number of nodes of $\mathcal{T}_k$ in Ω. δ_{ij} is the Kronecker symbol.

The functions $q_{l-1}^{(i)}$, $i = N_{l-1}+1, \ldots, N_l$, of (6) are continuous and piecewise *quadratic* in each triangle of $\mathcal{T}_{l-1}$. Again, they satisfy

$$q_{l-1}^{(i)}(x^{(j)}) = \delta_{ij} \quad \text{for } i,j = N_{l-1}+1, \ldots, N_l\,.$$

The basis of the space V_l^L we call *h–hierarchical basis* and the basis of the space V_l^Q is called *p–hierarchical basis.*

The finite element subspaces V_{l-1}^L, V_l^L, V_l^Q of (4), (5), and (6), respectively, give rise to the finite element stiffness matrices K_{l-1}^L, K_l^L, and K_l^Q as well as the load vectors $\underline{f}_{l-1}^L$, $\underline{f}_l^L$, and $\underline{f}_l^Q$.

For the computation of the coefficients of the element stiffness matrices and the element load vectors in general we must perform numerical integration. We therefore need an appropriate quadrature rule which guarantees the same FE discretization error as in the case of exact computation of the stiffness matrix and the load vector. To investigate the effect of numerical integration we will use well–known results as e.g. contained in [9]. For the sake of completeness we summarize some of them.

The application of quadrature rules for the computation of the matrix elements and the elements of the load vector results in an approximate bilinear form $\tilde{a}(\tilde{u}, \tilde{v})$ and an approximate right–hand side $\langle \tilde{F}, \tilde{v} \rangle$. Depending on the choice of the quadrature rule and the finite element subspace $\tilde{V}$, i.e. $\tilde{V} = V_{l-1}^L, \tilde{V} = V_l^L$, or $\tilde{V} = V_l^Q$, we will later describe $\tilde{a}(\tilde{u}, \tilde{v})$ in detail.

The approximate bilinear form is called *uniformly* $\tilde{V}$-elliptic, if there exists a constant $\tilde{\alpha} > 0$, $\tilde{\alpha}$ independent of $\tilde{V}$, such that

$$\tilde{a}(\tilde{v}, \tilde{v}) \geq \tilde{\alpha} ||\tilde{v}||^2_{1,2,\Omega} \quad \text{for all } \tilde{v} \in \tilde{V}.$$

Here $|| \cdot ||_{1,2,\Omega}$ denotes the norm in the Sobolev space $H^1(\Omega)$.

Using numerical integration, the boundary value problem (1) is approximated by

$$\text{Find } \tilde{u} \in \tilde{V} \text{ such that } \quad \tilde{a}(\tilde{u}, \tilde{v}) = \langle \tilde{F}, \tilde{v} \rangle \quad \text{for all } \tilde{v} \in \tilde{V}. \tag{7}$$

Theorem 1. (First Lemma of Strang) *Let the approximate bilinear form $\tilde{a}$ of (7) be uniformly $\tilde{V}$-elliptic. Then*

$$||u - \tilde{u}||_{1,2,\Omega} \leq c \left(\inf_{\tilde{v} \in \tilde{V}} \left\{ ||u - \tilde{v}||_{1,2,\Omega} + \sup_{\tilde{w} \in \tilde{V}} \frac{|a(\tilde{v}, \tilde{w}) - \tilde{a}(\tilde{v}, \tilde{w})|}{||\tilde{w}||_{1,2,\Omega}} \right\} + \sup_{\tilde{w} \in \tilde{V}} \frac{|\langle F, \tilde{w} \rangle - \langle \tilde{F}, \tilde{w} \rangle|}{||\tilde{w}||_{1,2,\Omega}} \right)$$

with a constant c which does not depend on the space $\tilde{V}$.

Let the solution $u \in H_0^{s+1}(\Omega)$, $a_{ij} \in W_\infty^s(\Omega)$, $i, j = 1, 2$, $f \in W_q^s(\Omega)$ with $q \geq 2$ and $q > 2/s$, and let the FE subspace $\tilde{V}$ contain piecewise polynomials of degree s, i.e. polynomials of degree s on the triangles of the triangulation. Furthermore, let the quadrature rule be exact for polynomial of degree $2s - 2$ on each triangle. Then the following estimate holds (see also [9])

$$\|u - \tilde{u}\|_{1,2,\Omega} \leq c h^s \left(|u|_{s+1,2,\Omega} + \sum_{i,j=1}^{2} \|a_{ij}\|_{s,\infty,\Omega} \|u\|_{s+1,2,\Omega} + \|f\|_{s,q,\Omega} \right).$$

Here $\| \cdot \|_{s+1,2,\Omega}$ and $| \cdot |_{s+1,2,\Omega}$ denote norms in $H_0^{s+1}(\Omega)$ as well as $\| \cdot \|_{s,q,\Omega}$ is a norm in $W_q^s(\Omega)$.

A multigrid algorithm with implicit extrapolation step

In Jung/Rüde [8] we have studied the convergence properties of a multigrid algorithm with implicit extrapolation step. However, the papers [8] were restricted to problems with piecewise constant functions $a_{ij}(x)$ and $f(x)$ in $\mathcal{T}_{l-1}$. If such a problem is discretized by *linear* elements, and the multigrid algorithm is combined with (implicit) extrapolation, the iterates converge to the solution given by *quadratic* elements. In this paper we will generalize this result to the case of variable coefficients. It will be shown that the extrapolation algorithm converges to the solution obtained with quadratic elements. In the analysis of this more general case, we will use special nonstandard quadrature rules.

In the following we will give a brief description of the smoothing procedure and the restriction operator used. Then we formulate the multigrid algorithm and study the convergence behavior.

Numbering the nodes in $\mathcal{T}_l$ such that the nodes which are also in the coarse mesh $\mathcal{T}_{l-1}$ appear first, we induce a block partitioning of the stiffness matrices

$$K_l^L = \begin{pmatrix} K_{l,vv}^L & K_{l,vm}^L \\ K_{l,mv}^L & K_{l,mm}^L \end{pmatrix}, \quad K_l^Q = \begin{pmatrix} K_{l,vv}^Q & K_{l,vm}^Q \\ K_{l,mv}^Q & K_{l,mm}^Q \end{pmatrix}. \tag{8}$$

In the multigrid algorithm we use the following smoothing procedures:

- Pre–smoothing $G_l^V(\underline{u}_l^{(j)}, K_l^L, \underline{f}_l^L)$: Let the initial guess $\underline{u}_l^{(j)} = (\underline{u}_{l,v}^{(j)}, \underline{u}_{l,m}^{(j)})^T$ be given. Set $\underline{u}_{l,v}^{(j+1)} = \underline{u}_{l,v}^{(j)}$ and compute an approximate solution $\underline{z}_{l,m}$ of the system

$$K_{l,mm}^L \underline{z}_{l,m} = \underline{f}_{l,m}^L - K_{l,mv}^L \underline{u}_{l,v}^{(j+1)} - K_{l,mm}^L \underline{u}_{l,m}^{(j)} \tag{9}$$

  by means of a linear iterative method starting with the zero vector. We suppose that the error transmission operator of the method is of the type

$$M_{l,m} = I_{l,m} - B_{l,mm}^{-1} K_{l,mm}^L.$$

  Then set $\underline{u}_l^{(j+1)} = (\underline{u}_{l,v}^{(j+1)}, \underline{u}_{l,m}^{(j)} + \underline{z}_{l,m})^T$.

- Post–smoothing $G_l^N(\underline{u}_l^{(j)}, K_l^L, \underline{f}_l^L)$: We use the same form of algorithm as for pre–smoothing. However, we suppose that the error transmission operator of the iterative method is of the form $M_{l,m} = I_{l,m} - B_{l,mm}^{-T} K_{l,mm}^L$ such that the overall multigrid operator becomes symmetric.

- We need the *injection operator*

$$I_l^{l-1,inj} : \mathrm{R}^{N_l} \longrightarrow \mathrm{R}^{N_{l-1}}$$

  in our algorithm.

Algorithm MG–EX

Let an initial guess $\underline{u}_l^{(k,0)}$ be given.

1. Pre–smoothing:

$$\underline{u}_l^{(k,1)} = G_l^V(\underline{u}_l^{(k,0)}, K_l^L, \underline{f}_l^L). \tag{10}$$

2. Coarse grid correction:

   (a) Compute the defect

$$\underline{d}_{l-1}^{(k)} = \frac{4}{3}\left(\underline{f}_{l,v}^L - K_{l,vv}^L \underline{u}_{l,v}^{(k,1)} - K_{l,vm}^L \underline{u}_{l,m}^{(k,1)}\right) - \frac{1}{3}\left(\underline{f}_{l-1}^L - K_{l-1}^L I_l^{l-1,inj} \underline{u}_l^{(k,1)}\right) \tag{11}$$

   (b) Solve

$$K_{l-1}^L \underline{w}_{l-1}^{(k)} = \underline{d}_{l-1}^{(k)}; \tag{12}$$

   using μ iteration steps of a usual symmetric multigrid ($(l-1)$–grid) algorithm, starting with the 0 vector and returning an approximate solution $\underline{\tilde{w}}_{l-1}^{(k)}$.

(c) Correct

$$\underline{u}_l^{(k,2)} = (\underline{u}_{l,v}^{(k,1)} + \underline{\tilde{w}}_{l-1}^{(k)}, \underline{u}_{l,m}^{(k,1)})^T \tag{13}$$

3. Post–smoothing:

$$\underline{u}_l^{(k,3)} = G_l^N(\underline{u}_l^{(k,2)}, K_l^L, \underline{f}_l^L) \tag{14}$$

and set $\underline{u}_l^{(k+1,0)} = \underline{u}_l^{(k,3)}$

Taking into consideration the definition of the smoothing procedures and the equivalence of step 2(a) to

$$\underline{d}_{l-1}^{(k)} = \left(\frac{4}{3}\underline{f}_{l,v}^L - \frac{1}{3}\underline{f}_{l-1}^L\right) - \left(\frac{4}{3}K_{l,vv}^L - \frac{1}{3}K_{l-1}^L\right)\underline{u}_{l,v}^{(k,1)} - \frac{4}{3}K_{l,vm}^L\underline{u}_{l,m}^{(k,1)}, \tag{15}$$

we can interpret our algorithm as a usual multigrid algorithm in the h–hierarchical basis to solve the system of equations

$$K_l^{L,ex}\underline{u}_l = \underline{f}_l^{L,ex} \tag{16}$$

with

$$K_l^{L,ex} = \begin{pmatrix} \frac{4}{3}K_{l,vv}^L - \frac{1}{3}K_{l-1}^L & \frac{4}{3}K_{l,vm}^L \\ \frac{4}{3}K_{l,mv}^L & \frac{4}{3}K_{l,mm}^L \end{pmatrix} \quad \text{and} \quad \underline{f}_l^{L,ex} = \begin{pmatrix} \frac{4}{3}\underline{f}_{l,v}^L - \frac{1}{3}\underline{f}_{l-1}^L \\ \frac{4}{3}\underline{f}_{l,m}^L \end{pmatrix}. \tag{17}$$

The main result of this section is that the iterates of the algorithm MG–EX converge to a FE solution which has the same order of discretization error as a FE solution obtained by p–hierarchical FE functions ($p = 2$).

Before we prove this fact, we introduce the quadrature rules that are used to compute the stiffness matrices and load vectors.

To obtain the entries of the stiffness matrices K_{l-1}^L, K_l^L, and K_l^Q, respectively, we have to compute

$$a(\tilde{p}_l^{(j)}, \tilde{p}_l^{(i)}) = \int_\Omega \left(A(x)\nabla_x\tilde{p}_l^{(j)}(x), \nabla_x\tilde{p}_l^{(i)}(x)\right) dx\,, \tag{18}$$

where $\tilde{p}_l^{(i)}$, $\tilde{p}_l^{(j)}$ stand for the functions $p_{l-1}^{(i)}$, $p_{l-1}^{(j)}$, $i,j = 1,\ldots,N_{l-1}$, $p_l^{(i)}$, $p_l^{(j)}$, $i,j = N_{l-1}+1,\ldots,N_l$, in the case of the h–hierarchical basis. In the case of the p–hierarchical basis the functions $\tilde{p}_l^{(i)}$, $\tilde{p}_l^{(j)}$ stand for $p_{l-1}^{(i)}$, $p_{l-1}^{(j)}$, $i,j = 1,\ldots,N_{l-1}$, $q_{l-1}^{(i)}$, $q_{l-1}^{(j)}$, $i,j = N_{l-1}+1,\ldots,N_l$.

First we explain the quadrature rules used for the computation of the matrices K_{l-1}^L and K_l^L. From (18) we obtain for the entries of K_{l-1}^L

$$\int_\Omega \left(A(x)\nabla_x p_{l-1}^{(j)}(x), \nabla_x p_{l-1}^{(i)}(x)\right) dx = \sum_{r\in\omega_{l-1}^{(ij)}} \int_{\delta_{l-1}^{(r)}} \left(A(x)\nabla_x p_{l-1}^{(j)}(x), \nabla_x p_{l-1}^{(i)}(x)\right) dx\,, \tag{19}$$

where

$$\omega_{l-1}^{(ij)} = \left\{r \,:\, p_{l-1}^{(i)} \not\equiv 0 \text{ and } p_{l-1}^{(j)} \not\equiv 0 \text{ on } \delta_{l-1}^{(r)}\right\}. \tag{20}$$

We transform the integrals over $\delta_{l-1}^{(r)}$ into integrals over the reference element $\Delta = \{(\xi_1, \xi_2) : 0 \le \xi_1 \le 1,\ 0 \le \xi_2 \le 1,\ \xi_1 + \xi_2 \le 1\}$. This leads to

$$\begin{aligned}
&\int_{\delta_{l-1}^{(r)}} \left(A(x)\nabla_x p_{l-1}^{(j)}(x), \nabla_x p_{l-1}^{(i)}(x)\right) dx \\
&= \int_{\Delta} \left(A(x)(J_{l-1}^{(r)})^{-T}\nabla_\xi p_{l-1}^{(j)}(x(\xi)), (J_{l-1}^{(r)})^{-T}\nabla_\xi p_{l-1}^{(i)}(x(\xi))\right) |\det J_{l-1}^{(r)}|\, d\xi \\
&= \int_{\Delta} \left(\bar{B}^{(r)}(x)\nabla_\xi \varphi_{\beta^{(r)}}(\xi), \nabla_\xi \varphi_{\alpha^{(r)}}(\xi)\right) d\xi \qquad (21)
\end{aligned}$$

with $\bar{B}^{(r)}(x) = (J_{l-1}^{(r)})^{-1} A(x) (J_{l-1}^{(r)})^{-T} |\det J_{l-1}^{(r)}|$ and $J_{l-1}^{(r)}$ from the transformation

$$\begin{aligned}
\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} &= \begin{pmatrix} x_1^{(r,2)} - x_1^{(r,1)} & x_1^{(r,3)} - x_1^{(r,1)} \\ x_2^{(r,2)} - x_2^{(r,1)} & x_2^{(r,3)} - x_2^{(r,1)} \end{pmatrix} \begin{pmatrix} \xi_1 \\ \xi_2 \end{pmatrix} + \begin{pmatrix} x_1^{(r,1)} \\ x_2^{(r,1)} \end{pmatrix} \\
&= J_{l-1}^{(r)} \begin{pmatrix} \xi_1 \\ \xi_2 \end{pmatrix} + \begin{pmatrix} x_1^{(r,1)} \\ x_2^{(r,1)} \end{pmatrix}
\end{aligned} \qquad (22)$$

Here $x_i^{(r,\alpha)}$, $i, j = 1, 2$, $\alpha = 1, 2, 3$, denotes the coordinates of the vertices of the triangle $\delta_{l-1}^{(r)}$, and $\alpha^{(r)}$ as well as $\beta^{(r)}$ are the local numbers of the vertices $P^{(i)}$ and $P^{(j)}$. The linear functions $\varphi_{\alpha^{(r)}}$, $\varphi_{\beta^{(r)}}$, $\alpha^{(r)}$, $\beta^{(r)} = 1, 2, 3$, on the reference element are defined by

$$\varphi_1(\xi) = 1 - \xi_1 - \xi_2\,, \quad \varphi_2(\xi) = \xi_1\,, \quad \text{and} \quad \varphi_3(\xi) = \xi_2. \qquad (23)$$

The following equivalent formulation of (21) is the basis of the application of our quadrature rules.

With the directional derivative

$$\frac{\partial \varphi}{\partial \xi_s} = \frac{\partial \varphi}{\partial \xi_2} - \frac{\partial \varphi}{\partial \xi_1} \qquad (24)$$

we obtain

$$\begin{aligned}
&\int_{\Delta} \left(\bar{b}_{11}^{(r)} \frac{\partial \varphi_{\beta^{(r)}}}{\partial \xi_1} \frac{\partial \varphi_{\alpha^{(r)}}}{\partial \xi_1} + \bar{b}_{12}^{(r)} \left(\frac{\partial \varphi_{\beta^{(r)}}}{\partial \xi_1} \frac{\partial \varphi_{\alpha^{(r)}}}{\partial \xi_2} + \frac{\partial \varphi_{\beta^{(r)}}}{\partial \xi_2} \frac{\partial \varphi_{\alpha^{(r)}}}{\partial \xi_1} \right) + \bar{b}_{22}^{(r)} \frac{\partial \varphi_{\beta^{(r)}}}{\partial \xi_2} \frac{\partial \varphi_{\alpha^{(r)}}}{\partial \xi_2} \right) d\xi \\
&= \int_{\Delta} \left(b_{11}^{(r)} \frac{\partial \varphi_{\beta^{(r)}}}{\partial \xi_1} \frac{\partial \varphi_{\alpha^{(r)}}}{\partial \xi_1} + b_{22}^{(r)} \frac{\partial \varphi_{\beta^{(r)}}}{\partial \xi_2} \frac{\partial \varphi_{\alpha^{(r)}}}{\partial \xi_2} + b_{12}^{(r)} \frac{\partial \varphi_{\beta^{(r)}}}{\partial \xi_s} \frac{\partial \varphi_{\alpha^{(r)}}}{\partial \xi_s} \right) d\xi, \qquad (25)
\end{aligned}$$

where

$$\begin{aligned}
b_{11}^{(r)}(x(\xi)) &= \bar{b}_{11}^{(r)}(x(\xi)) + \bar{b}_{12}^{(r)}(x(\xi))\,, \qquad b_{22}^{(r)}(x(\xi)) = \bar{b}_{22}^{(r)}(x(\xi)) + \bar{b}_{12}^{(r)}(x(\xi))\,, \\
b_{12}^{(r)}(x(\xi)) &= -\bar{b}_{12}^{(r)}(x(\xi))\,.
\end{aligned}$$

For the numerical integration of the three terms in (25) we use the following three quadrature rules

$$\int_{\Delta} v(\xi)d\xi = \text{meas}_{\Delta}\, v(\xi^{(\sigma)}), \qquad \sigma = 1,2,3, \tag{26}$$

with

$$\xi^{(1)} = \left(\frac{1}{2},0\right),\ \xi^{(2)} = \left(0,\frac{1}{2}\right),\ \text{and}\ \xi^{(3)} = \left(\frac{1}{2},\frac{1}{2}\right), \tag{27}$$

respectively. Obviously the quadrature rules in (26) are exact for constant functions v.

The elements of the matrix K_l^L are computed in the same way. We can write the expressions for the computation of the matrix elements in the following formulation

$$\begin{aligned} a(\tilde{p}_l^{(j)},\tilde{p}_l^{(i)}) &= \sum_{r\in\omega_{l-1}^{(ij)}} \int_{\delta_{l-1}^{(r)}} \left(A(x)\nabla_x \tilde{p}_l^{(j)}(x), \nabla_x \tilde{p}_l^{(i)}(x)\right) dx \\ &= \sum_{r\in\omega_{l-1}^{(ij)}} \int_{\Delta} \left(A(x)(J_{l-1}^{(r)})^{-T}\nabla_\xi \tilde{p}_l^{(j)}(x(\xi)), (J_{l-1}^{(r)})^{-T}\nabla_\xi \tilde{p}_l^{(i)}(x(\xi))\right) |\det J_{l-1}^{(r)}|\, d\xi \\ &= \sum_{r\in\omega_{l-1}^{(ij)}} \sum_{k=1}^{4} \int_{\Delta^{(k)}} \left(\bar{B}^{(r)}(x)\nabla_\xi \varphi_{\beta^{(r)}}(\xi), \nabla_\xi \varphi_{\alpha^{(r)}}(\xi)\right) d\xi\,, \end{aligned} \tag{28}$$

where again $\alpha^{(r)}, \beta^{(r)} = 1,2,\ldots,6$, are the local numbers of the nodes $P^{(i)}$ and $P^{(j)}$, $\omega_{l-1}^{(ij)} = \left\{r\ :\ \tilde{p}_l^{(i)} \not\equiv 0 \ \text{ and } \ \tilde{p}_l^{(j)} \not\equiv 0 \ \text{ on } \ \delta_{l-1}^{(r)}\right\}$, $\Delta = \cup_{k=1}^4 \Delta^{(k)}$ (see also Figure 1), and

$$\begin{aligned} \varphi_1(\xi) &= 1-\xi_1-\xi_2\,, \\ \varphi_2(\xi) &= \xi_1\,, \\ \varphi_3(\xi) &= \xi_2\,, \end{aligned} \qquad \varphi_4(\xi) = \begin{cases} 2\xi_1 & \text{in } \Delta^{(1)} \\ 2-2\xi_1-2\xi_2 & \text{in } \Delta^{(2)} \\ 0 & \text{in } \Delta^{(3)} \\ 1-2\xi_2 & \text{in } \Delta^{(4)} \end{cases}, \tag{29}$$

$$\varphi_5(\xi) = \begin{cases} 0 & \text{in } \Delta^{(1)} \\ 2\xi_2 & \text{in } \Delta^{(2)} \\ 2\xi_1 & \text{in } \Delta^{(3)} \\ 2\xi_1+2\xi_2-1 & \text{in } \Delta^{(4)} \end{cases} \qquad \varphi_6(\xi) = \begin{cases} 2\xi_2 & \text{in } \Delta^{(1)} \\ 0 & \text{in } \Delta^{(2)} \\ 2-2\xi_1-2\xi_2 & \text{in } \Delta^{(3)} \\ 1-2\xi_1 & \text{in } \Delta^{(4)} \end{cases}$$

To compute each integral over $\Delta^{(k)}$ in (28) we use the equivalent formulation of type (25) and a quadrature rule of type (26).

In the case of the p–hierarchical basis, we have to compute the entries of the matrix K_l^Q, i.e. expressions of the form (18), where $\tilde{p}_l^{(i)}, \tilde{p}_l^{(j)}$ stand for the functions $p_{l-1}^{(i)}, p_{l-1}^{(j)}$, $i,j = 1,\ldots,N_{l-1}$, $q_{l-1}^{(i)}, q_{l-1}^{(j)}$, $i,j = N_{l-1}+1,\ldots,N_l$.

Again we get

$$\int_{\Omega} \left(A(x)\nabla_x \tilde{p}_l^{(j)}(x), \nabla_x \tilde{p}_l^{(i)}(x)\right) dx = \sum_{r\in\omega_{l-1}^{(ij)}} \int_{\delta_{l-1}^{(r)}} \left(A(x)\nabla_x \tilde{p}_l^{(j)}(x), \nabla_x \tilde{p}_l^{(i)}(x)\right) dx\,, \tag{30}$$

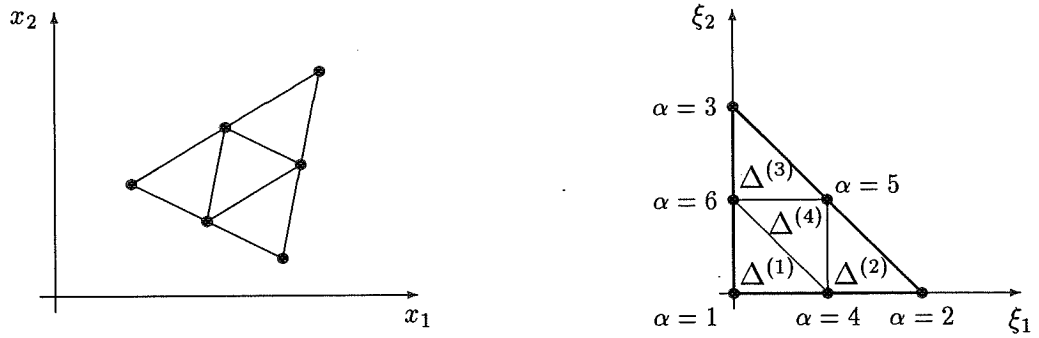


Figure 1: An arbitrary triangle $\delta_{l-1}^{(r)}$ and the reference element Δ

with $\omega_{l-1}^{(ij)}$ from (20). After the transformation of the integrals over $\delta_{l-1}^{(r)}$ into integrals over the reference element Δ we obtain the integrals

$$\int_{\Delta} \left(\bar{B}^{(r)}(x) \nabla_{\xi} \psi_{\beta^{(r)}}(\xi), \nabla_{\xi} \psi_{\alpha^{(r)}}(\xi) \right) d\xi \,. \tag{31}$$

The functions $\psi_{\alpha^{(r)}}$ and $\psi_{\beta^{(r)}}$, $\alpha^{(r)}$, $\beta^{(r)} = 1, 2, \ldots, 6$, are defined by

$$\begin{aligned} &\psi_1(\xi) = 1 - \xi_1 - \xi_2 \,, && \psi_2(\xi) = \xi_1 \,, && \psi_3(\xi) = \xi_2 \,, \\ &\psi_4(\xi) = 4\xi_1(1 - \xi_1 - \xi_2) \,, && \psi_5(\xi) = 4\xi_1\xi_2 \,, && \psi_6(\xi) = 4\xi_2(1 - \xi_1 - \xi_2) \,. \end{aligned} \tag{32}$$

The integral (31) we write in the form (25). For the numerical integration of the resulting integrals over Δ we use quadrature rules, which we derive from the quadrature rules (26) by extrapolation. Specifically, we apply for the computation of the first, the second, and the third term the quadrature rules

$$\int_{\Delta} v(\xi) d\xi \approx \left\{ \frac{4}{3} \left(\frac{1}{4} v(\xi^{(4)}) + \frac{1}{4} v(\xi^{(5)}) + \frac{1}{2} v(\xi^{(6)}) \right) - \frac{1}{3} v(\xi^{(1)}) \right\} \text{meas}\, \Delta \tag{33}$$

$$\int_{\Delta} v(\xi) d\xi \approx \left\{ \frac{4}{3} \left(\frac{1}{4} v(\xi^{(7)}) + \frac{1}{4} v(\xi^{(8)}) + \frac{1}{2} v(\xi^{(9)}) \right) - \frac{1}{3} v(\xi^{(2)}) \right\} \text{meas}\, \Delta \tag{34}$$

$$\int_{\Delta} v(\xi) d\xi \approx \left\{ \frac{4}{3} \left(\frac{1}{4} v(\xi^{(10)}) + \frac{1}{4} v(\xi^{(11)}) + \frac{1}{2} v(\xi^{(12)}) \right) - \frac{1}{3} v(\xi^{(3)}) \right\} \text{meas}\, \Delta \tag{35}$$

with $\xi^{(1)}$, $\xi^{(2)}$, $\xi^{(3)}$ from (27) and

$$\begin{aligned} \xi^{(4)} &= (\tfrac{1}{4}, 0) \,, & \xi^{(5)} &= (\tfrac{3}{4}, 0) \,, & \xi^{(6)} &= (\tfrac{1}{4}, \tfrac{1}{2}) \,, \\ \xi^{(7)} &= (0, \tfrac{1}{4}) \,, & \xi^{(8)} &= (0, \tfrac{3}{4}) \,, & \xi^{(9)} &= (\tfrac{1}{2}, \tfrac{1}{4}) \,, \\ \xi^{(10)} &= (\tfrac{3}{4}, \tfrac{1}{4}) \,, & \xi^{(11)} &= (\tfrac{1}{4}, \tfrac{3}{4}) \,, & \xi^{(12)} &= (\tfrac{1}{4}, \tfrac{1}{4}) \,. \end{aligned} \tag{36}$$

A simple calculation shows that the quadrature rules (33)–(35) are exact for quadratic functions.

Because of the smoothness of the coefficient functions a_{ij} in (2) one can prove that the quadrature rules (26) and (33)–(35) lead for sufficiently small discretization parameters h to a uniformly $\tilde{V}$–elliptic bilinear form $\tilde{a}(.,.)$.

In the following we prove that the extrapolated stiffness matrix in (17) is equal to the stiffness matrix resulting from a discretization with p–hierarchical functions, where we assume that we use the quadrature rules (26) in case of the h-hierarchical basis, and (33–35) in case of the p-hierarchical basis.

Lemma 2. *If we compute the element stiffness matrices K_{l-1}^L, K_l^L, and K_l^Q as described above, i.e. by means of the quadrature rules (26) and (33) – (35), the relation*

$$K_l^{L,ex} = K_l^Q \tag{37}$$

holds.

Proof: The proof is based on comparing the matrices $K_l^{L,ex}$ and K_l^Q element by element. The extrapolated stiffness matrix $K_l^{L,ex}$ and the matrix K_l^Q have the block structure

$$K_l^{L,ex} = \begin{pmatrix} \frac{4}{3}K_{l,vv}^L - \frac{1}{3}K_{l-1}^L & \frac{4}{3}K_{l,vm}^L \\ \frac{4}{3}K_{l,mv}^L & \frac{4}{3}K_{l,mm}^L \end{pmatrix}, \quad K_l^Q = \begin{pmatrix} K_{l,vv}^Q & K_{l,vm}^Q \\ K_{l,mv}^Q & K_{l,mm}^Q \end{pmatrix} \tag{38}$$

The entries of the stiffness matrix K_{l-1}^L are computed using relations (18)–(23) and for the computation of the elements of the matrix K_l^L we use relations (28)–(29).

First, we now prove the identity of the coarse mesh blocks $K_{l,vv}^{L,ex} = K_{l,vv}^Q$. Using the quadrature rules of type (26) and the representation (25) with $\alpha^{(r)}$, $\beta^{(r)} = 1, 2, 3$, the elements of the matrix $K_{l,vv}^{L,ex}$ are defined by

$$\begin{aligned} K_{l,vv}^{L,ex,(ij)} = \sum_{r\in\omega_{l-1}^{(ij)}} \Bigg\{ & \\ & \frac{4}{3}\sum_{t=4}^{6}\gamma_t^L b_{11}^{(r)}(x(\xi^{(t)}))\frac{\partial\varphi_{\beta^{(r)}}(\xi^{(t)})}{\partial\xi_1}\frac{\partial\varphi_{\alpha^{(r)}}(\xi^{(t)})}{\partial\xi_1} - \frac{1}{3}\gamma_1^L b_{11}^{(r)}(x(\xi^{(1)}))\frac{\partial\varphi_{\beta^{(r)}}(\xi^{(1)})}{\partial\xi_1}\frac{\partial\varphi_{\alpha^{(r)}}(\xi^{(1)})}{\partial\xi_1} \\ & + \frac{4}{3}\sum_{t=7}^{9}\gamma_t^L b_{22}^{(r)}(x(\xi^{(t)}))\frac{\partial\varphi_{\beta^{(r)}}(\xi^{(t)})}{\partial\xi_2}\frac{\partial\varphi_{\alpha^{(r)}}(\xi^{(t)})}{\partial\xi_2} - \frac{1}{3}\gamma_2^L b_{22}^{(r)}(x(\xi^{(2)}))\frac{\partial\varphi_{\beta^{(r)}}(\xi^{(2)})}{\partial\xi_2}\frac{\partial\varphi_{\alpha^{(r)}}(\xi^{(2)})}{\partial\xi_2} \\ & + \frac{4}{3}\sum_{t=10}^{12}\gamma_t^L b_{12}^{(r)}(x(\xi^{(t)}))\frac{\partial\varphi_{\beta^{(r)}}(\xi^{(t)})}{\partial\xi_s}\frac{\partial\varphi_{\alpha^{(r)}}(\xi^{(t)})}{\partial\xi_s} - \frac{1}{3}\gamma_3^L b_{12}^{(r)}(x(\xi^{(3)}))\frac{\partial\varphi_{\beta^{(r)}}(\xi^{(3)})}{\partial\xi_s}\frac{\partial\varphi_{\alpha^{(r)}}(\xi^{(3)})}{\partial\xi_s} \Bigg\} \end{aligned}$$

with $\gamma_1^L = \gamma_2^L = \gamma_3^L = \operatorname{meas}\Delta$, $\gamma_4^L = \gamma_5^L = \gamma_7^L = \gamma_8^L = \gamma_{10}^L = \gamma_{11}^L = \operatorname{meas}\Delta^{(k)}$, and $\gamma_6^L = \gamma_9^L = \gamma_{12}^L = 2\operatorname{meas}\Delta^{(k)}$.

For the entries of the matrix K_l^Q we get by using relations (30)–(32) and the quadrature rules (33)–(35)

$$K_{l,vv}^{Q,(ij)} = \sum_{r\in\omega_{l-1}^{(ij)}} \Bigg\{$$

$$\frac{4}{3}\sum_{t=4}^{6}\gamma_t^Q b_{11}^{(r)}(x(\xi^{(t)}))\frac{\partial\psi_{\beta(r)}(\xi^{(t)})}{\partial\xi_1}\frac{\partial\psi_{\alpha(r)}(\xi^{(t)})}{\partial\xi_1} - \frac{1}{3}\gamma_1^Q b_{11}^{(r)}(x(\xi^{(1)}))\frac{\partial\psi_{\beta(r)}(\xi^{(1)})}{\partial\xi_1}\frac{\partial\psi_{\alpha(r)}(\xi^{(1)})}{\partial\xi_1}$$

$$+\frac{4}{3}\sum_{t=7}^{9}\gamma_t^Q b_{22}^{(r)}(x(\xi^{(t)}))\frac{\partial\psi_{\beta(r)}(\xi^{(t)})}{\partial\xi_2}\frac{\partial\psi_{\alpha(r)}(\xi^{(t)})}{\partial\xi_2} - \frac{1}{3}\gamma_2^Q b_{22}^{(r)}(x(\xi^{(2)}))\frac{\partial\psi_{\beta(r)}(\xi^{(2)})}{\partial\xi_2}\frac{\partial\psi_{\alpha(r)}(\xi^{(2)})}{\partial\xi_2}$$

$$+\frac{4}{3}\sum_{t=10}^{12}\gamma_t^Q b_{12}^{(r)}(x(\xi^{(t)}))\frac{\partial\psi_{\beta(r)}(\xi^{(t)})}{\partial\xi_s}\frac{\partial\psi_{\alpha(r)}(\xi^{(t)})}{\partial\xi_s} - \frac{1}{3}\gamma_3^Q b_{12}^{(r)}(x(\xi^{(3)}))\frac{\partial\psi_{\beta(r)}(\xi^{(3)})}{\partial\xi_s}\frac{\partial\psi_{\alpha(r)}(\xi^{(3)})}{\partial\xi_s}\Bigg\}$$

with $\gamma_1^Q = \gamma_2^Q = \gamma_3^Q = \text{meas}\,\Delta$, $\gamma_4^Q = \gamma_5^Q = \gamma_7^Q = \gamma_8^Q = \gamma_{10}^Q = \gamma_{11}^Q = 0.25\,\text{meas}\,\Delta$ and $\gamma_6^Q = \gamma_9^Q = \gamma_{12}^Q = 0.5\,\text{meas}\,\Delta$.

	$\xi^{(1)}$	$\xi^{(4)}$	$\xi^{(5)}$	$\xi^{(6)}$		$\xi^{(2)}$	$\xi^{(7)}$	$\xi^{(8)}$	$\xi^{(9)}$		$\xi^{(3)}$	$\xi^{(10)}$	$\xi^{(11)}$	$\xi^{(12)}$
$\frac{\partial\varphi_1}{\partial\xi_1}$	-1	-1	-1	-1	$\frac{\partial\varphi_1}{\partial\xi_2}$	-1	-1	-1	-1	$\frac{\partial\varphi_1}{\partial\xi_s}$	0	0	0	0
$\frac{\partial\varphi_2}{\partial\xi_1}$	1	1	1	1	$\frac{\partial\varphi_2}{\partial\xi_2}$	0	0	0	0	$\frac{\partial\varphi_2}{\partial\xi_s}$	-1	-1	-1	-1
$\frac{\partial\varphi_3}{\partial\xi_1}$	0	0	0	0	$\frac{\partial\varphi_3}{\partial\xi_2}$	1	1	1	1	$\frac{\partial\varphi_3}{\partial\xi_s}$	1	1	1	1
$\frac{\partial\varphi_4}{\partial\xi_1}$	—	2	-2	0	$\frac{\partial\varphi_4}{\partial\xi_2}$	—	0	0	-2	$\frac{\partial\varphi_4}{\partial\xi_s}$	—	0	0	-2
$\frac{\partial\varphi_5}{\partial\xi_1}$	—	0	0	2	$\frac{\partial\varphi_5}{\partial\xi_2}$	—	0	0	2	$\frac{\partial\varphi_5}{\partial\xi_s}$	—	2	-2	0
$\frac{\partial\varphi_6}{\partial\xi_1}$	—	0	0	-2	$\frac{\partial\varphi_6}{\partial\xi_2}$	—	2	-2	0	$\frac{\partial\varphi_6}{\partial\xi_s}$	—	0	0	2
$\frac{\partial\psi_4}{\partial\xi_1}$	0	2	-2	0	$\frac{\partial\psi_4}{\partial\xi_2}$	0	0	0	-2	$\frac{\partial\psi_4}{\partial\xi_s}$	0	0	0	-2
$\frac{\partial\psi_5}{\partial\xi_1}$	0	0	0	2	$\frac{\partial\psi_5}{\partial\xi_2}$	0	0	0	2	$\frac{\partial\psi_5}{\partial\xi_s}$	0	2	-2	0
$\frac{\partial\psi_6}{\partial\xi_1}$	0	0	0	-2	$\frac{\partial\psi_6}{\partial\xi_2}$	0	2	-2	0	$\frac{\partial\psi_6}{\partial\xi_s}$	0	0	0	2

Table 1: Values of the partial derivatives in the quadrature points

The symbol "—" in Table 1 means that the partial derivative does not exist in this quadrature point. But we do not need these values for the computation of the matrix elements of $K_l^{L,ex}$

If we examine the values of the partial derivatives $\partial\varphi_{\alpha^{(r)}}/\partial\xi_1$, $\partial\varphi_{\alpha^{(r)}}/\partial\xi_2$, $\partial\varphi_{\alpha^{(r)}}/\partial\xi_s$, $\partial\psi_{\alpha^{(r)}}/\partial\xi_1$, $\partial\psi_{\alpha^{(r)}}/\partial\xi_2$, $\partial\psi_{\alpha^{(r)}}/\partial\xi_s$, $\alpha^{(r)} = 1,2,3$, given in Table 1 and the relations $\operatorname{meas}\Delta = 4\operatorname{meas}\Delta^{(k)}$, $k = 1,2,3,4$, $\varphi_{\alpha^{(r)}} = \psi_{\alpha^{(r)}}$, $\alpha^{(r)} = 1,2,3$, we see that

$$K_{l,vv}^{r,L,ex,(ij)} = K_{l,vv}^{r,Q,(ij)} \quad \text{for } i,j = 1,2,\ldots,N_{l-1}\,, \quad \text{i.e. } K_{l,vv}^{r,L} = K_{l,vv}^{r,Q}\,.$$

In the same way we can prove

$$K_{l,vm}^{r,L,ex} = K_{l,vm}^{r,Q}\,, \quad K_{l,mv}^{r,L,ex} = K_{l,mv}^{r,Q} \quad \text{and} \quad K_{l,mm}^{r,L,ex} = K_{l,mm}^{r,Q}\,.$$

This completes the proof. ∎

Next we discuss the computation of the entries of the load vectors. For the entries of the vector $\underline{f}_{l-1}^{r,L}$ we get

$$f_{l-1}^{r,L,(i)} = \langle F, p_{l-1}^{(i)}\rangle = \sum_{r\in\omega_{l-1}^{(i)}} \int_{\delta_{l-1}^{(r)}} f(x)p_{l-1}^{(i)}(x)\,dx = \sum_{r\in\omega_{l-1}^{(i)}} \int_{\Delta} f(x(\xi))\varphi_{\alpha^{(r)}}(\xi)|\det J_{l-1}^{(r)}|\,d\xi \tag{39}$$

with $\omega_{l-1}^{(i)} = \{r\,:\, p_{l-1}^{(i)} \not\equiv 0 \text{ on } \delta_{l-1}^{(r)}\}$.

The integrals over Δ we compute by using the quadrature rule

$$\int_{\Delta} v(\xi)\,d\xi = \left(\frac{1}{3}v(0,0) + \frac{1}{3}v(1,0) + \frac{1}{3}v(0,1)\right)\operatorname{meas}\Delta\,. \tag{40}$$

Obviously, this formula is exact for linear functions v.

The entries of the vector $\underline{f}_l^{r,L}$ are defined by

$$f_l^{r,L,(i)} = \langle F, \tilde{p}_l^{(i)}\rangle = \sum_{r\in\omega_{l-1}^{(i)}} \int_{\delta_{l-1}^{(r)}} f(x)\tilde{p}_l^{(i)}(x)\,dx = \sum_{r\in\omega_{l-1}^{(i)}} \int_{\Delta} f(x(\xi))\varphi_{\alpha^{(r)}}(\xi)|\det J_{l-1}^{(r)}|\,d\xi$$

$$= \sum_{r\in\omega_{l-1}^{(i)}} \sum_{k=1}^{4} \int_{\Delta^{(k)}} f(x(\xi))\varphi_{\alpha^{(r)}}(\xi)|\det J_{l-1}^{(r)}|\,d\xi \tag{41}$$

with $\tilde{p}_l^{(i)} = p_{l-1}^{(i)}$ for $i = 1,\ldots,N_{l-1}$, $\tilde{p}_l^{(i)} = p_l^{(i)}$ for $i = N_{l-1}+1,\ldots,N_l$ and the functions $\varphi_{\alpha^{(r)}}$ from (29). The integrals over $\Delta^{(k)}$ are computed by a formula of the type (40).

In the case of the p–hierarchical basis, the entries of the load vector $\underline{f}_l^{r,Q}$ are given by

$$f_l^{r,Q,(i)} = \langle F, \tilde{p}_l^{(i)}\rangle = \sum_{r\in\omega_{l-1}^{(i)}} \int_{\delta_{l-1}^{(r)}} f(x)\tilde{p}_l^{(i)}(x)\,dx = \sum_{r\in\omega_{l-1}^{(i)}} \int_{\Delta} f(x(\xi))\psi_{\alpha^{(r)}}(\xi)|\det J_{l-1}^{(r)}|\,d\xi \tag{42}$$

with $\tilde{p}_l^{(i)} = p_{l-1}^{(i)}$ for $i = 1,\ldots,N_{l-1}$, $\tilde{p}_l^{(i)} = q_{l-1}^{(i)}$ for $i = N_{l-1}+1,\ldots,N_l$, and the functions $\psi_{\alpha^{(r)}}$ from (32). For the computation of the integrals over Δ we use the quadrature rule

$$\int_{\Delta} v(\xi)\,d\xi = \left(\frac{1}{3}v(\xi^{(1)}) + \frac{1}{3}v(\xi^{(2)}) + \frac{1}{3}v(\xi^{(3)})\right)\operatorname{meas}\Delta\,. \tag{43}$$

with $\xi^{(\sigma)}$, $\sigma = 1, 2, 3$, from (27). This formula is exact for quadratic functions v.

Lemma 3. *If the load vectors* $\underline{f}_{l-1}^{L}$, $\underline{f}_{l}^{L}$, *and* $\underline{f}_{l}^{Q}$ *are defined as described above, then the relation*

$$\underline{f}_{l}^{L,ex} = \underline{f}_{l}^{Q} \tag{44}$$

holds.

Proof: Using the relations (39), (41) and quadrature rules of type (40) for computing the extrapolated load vector $\underline{f}_{l}^{L,ex}$ (see (17)) as well as relations (42) and (43) for computing the load vector $\underline{f}_{l}^{Q}$, the proof follows immediately. ∎

A consequence of Lemma 2 and Lemma 3 is the following Theorem.

Theorem 4. *If the extrapolated stiffness matrix* $K_{l}^{L,ex}$ *and the extrapolated load vector* $\underline{f}_{l}^{L,ex}$ *as well as the stiffness matrix* K_{l}^{Q} *and the load vector* $\underline{f}_{l}^{Q}$ *are computed as described above, then the systems of algebraic FE equations*

$$K_{l}^{L,ex}\underline{u}_{l} = \underline{f}_{l}^{L,ex} \qquad \textit{and} \qquad K_{l}^{Q}\underline{u}_{l} = \underline{f}_{l}^{Q} \tag{45}$$

have the same solution.

Now we can immediately prove the following convergence theorem for the algorithm MG–EX.

Theorem 5. *Under the assumption that the extrapolated stiffness matrix* $K_{l}^{L,ex}$, *the extrapolated load vector* $\underline{f}_{l}^{L,ex}$, *the stiffness matrix* K_{l}^{Q}, *and the load vector* $\underline{f}_{l}^{Q}$ *are computed as discussed in this section, the following statements hold.*

(i) The iterates of algorithm MG–EX converge to a FE solution which has the same discretization error as a FE solution obtained by a FE discretization with p–hierarchical functions.

(ii) The convergence rate of algorithm MG–EX does not depend on the discretization parameter.

Proof: The statement (i) follows from the interpretation of algorithm MG–EX as a usual multigrid algorithm for solving the system of algebraic equation $K_{l}^{L,ex}\underline{u}_{l} = \underline{f}_{l}^{L,ex}$and the equivalence of the systems of algebraic equations $K_{l}^{L,ex}\underline{u}_{l} = \underline{f}_{l}^{L,ex}$ and $K_{l}^{Q}\underline{u}_{l} = \underline{f}_{l}^{Q}$.

Statement (ii) we can prove in an analogous way as done for the piecewise constant coefficient case in [8]. ∎

Remark: We can also formulate algorithm MG–EX in terms of a piecewise linear nodal basis. All our results are also valid in this case.

Numerical results

In this Section we want to confirm our theoretical results by a numerical example. We will illustrate that the iterates of the algorithm MG–EX converge to the FE solution which we would obtain by a discretization of problem (1) with p–hierarchical functions. Furthermore, the numerical example shows that the convergence rate of algorithm MG–EX is independent of the discretization parameter.

All algorithms have been implemented within the multigrid package FEMGP [10]. The computations were performed on a PC 80486 (33 MHz) using the LAHEY–Fortran compiler.

Let us consider the problem (1), where $\Omega = (0,1) \times (0,1)$,

$$A = \begin{pmatrix} a_{11}(x) & 0 \\ 0 & a_{22}(x) \end{pmatrix}, \quad a_{11}(x) = (1.1 - \tanh(3x_1 + 3x_2 - 4.5)), \text{and} \quad a_{22}(x) = 2a_{11}(x).$$

The right–hand side $f(x)$ is chosen such that the function

$$u(x) = x_1(1 - x_1)x_2(1 - x_2)(1 + \tanh(3x_1 + 3x_2 - 4.5))$$

is the exact solution of problem (1).

Starting from the coarsest triangulation $\mathcal{T}_1$ (see Figure 2) the finer triangulations have been generated by dividing all triangles of the triangulation $\mathcal{T}_k$, $k = 1, 2, \ldots, l-1$, into four smaller congruent sub–triangles. In Table 2 we give the numbers of nodes and the numbers of triangles in each triangulation.

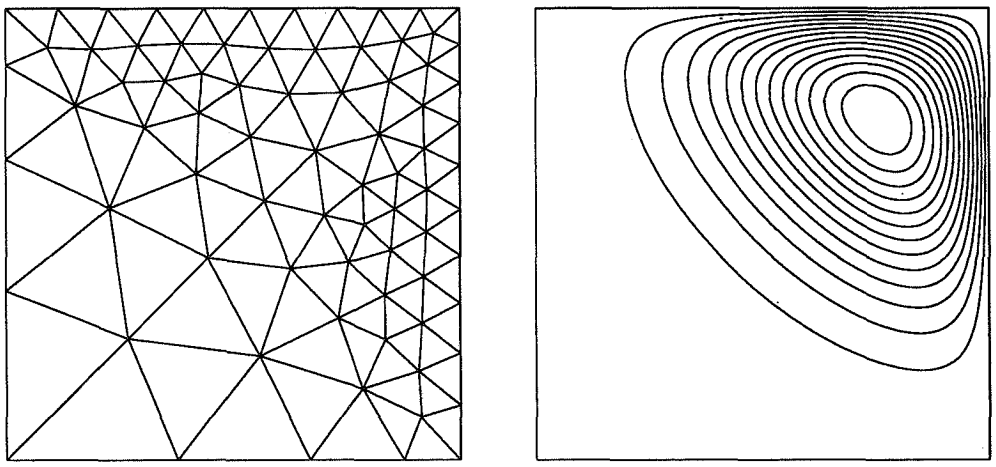


Figure 2: Mesh $\mathcal{T}_1$ and iso–lines of the solution u

For Algorithm MG–EX we used as pre–smoother two sweeps of the lexicographically forward Gauss–Seidel method for solving system (9), one iteration step of a $(l-1)$–grid algorithm for solving the coarse–grid system (12), and two sweeps of the

triangulation	$\mathcal{T}_1$	$\mathcal{T}_2$	$\mathcal{T}_3$	$\mathcal{T}_4$	$\mathcal{T}_5$
number of nodes	78	281	1065	4145	16353
number of triangles	126	504	2016	8064	32256

Table 2: Number of nodes and number of triangles in $\mathcal{T}_k$, $k = 1, 2, \ldots, 5$

lexicographically backward Gauss–Seidel method in the post–smoothing step. The initial guess was obtained by a full multigrid strategy. On the levels $k = 1, 2, \ldots, l-1$ a usual multigrid algorithm for solving the corresponding FE equations in the linear nodal basis was performed. Within this k–grid algorithms one V–cycle with two Gauss-Seidel sweeps lexicographically forward in the pre–smoothing step and two Gauss–Seidel sweeps lexicographically backward in the post–smoothing step were used. The convergence criterion for MG–EX was

$$\| \underline{f}_l^{L,ex} - K_l^{L,ex} \underline{u}_l^{(k+1,0)} \| \leq 10^{-4} \, \| \underline{f}_l^{L,ex} - K_l^{L,ex} \underline{u}_l^{(0,0)} \| \,, \tag{46}$$

where $\| \, . \, \|$ denotes the Euclidean norm in the space R^{N_l}, and $\underline{u}_l^{(0,0)}$ is the initial guess.

In Table 3 we present the number of iterations and the CPU–time needed by the application of the algorithm MG–EX. An improvement of the convergence behavior of our algorithm we obtain by introducing additional pre–smoothing and post–smoothing steps, i.e. before step 1 in the algorithm MG–EX we perform one iteration step of the Gauss–Seidel method lexicographically forward and after step 3 one iteration step of the Gauss-Seidel method lexicographically backward applied to the system of algebraic equations $K_l^{L,ex} \underline{u}_l^{L,ex} = \underline{f}_l^{L,ex}$. This is illustrated in column MG–EX(1) of Table 3.

l	Algorithm MG–EX		Algorithm MG–EX(1)	
	number of iterations	CPU–time	number of iterations	CPU–time
3	11	2.06 sec	6	1.54 sec
4	11	9.61 sec	5	6.28 sec
5	12	45.77 sec	5	27.54 sec

Table 3: Comparison of the algorithm MG–EX and the algorithm MG–EX(1)

Finally, we compare the discretization errors $\|u - u_l^{L,ex}\|$ and $\|u - u_l^Q\|$ in the H^1–norm and in the L_2–norm. Here $u_l^{L,ex}$ denotes the FE solution obtained by means of the algorithm MG–EX and u_l^Q the FE solution by a discretization with piecewise p–hierarchical functions.

Table 4 shows that the algorithm MG–EX yields discretization errors which are typical for discretizations with piecewise quadratic functions, i.e. we can observe an error of order $O(h_l^2)$ in the H^1–norm and $O(h_l^3)$ in the L_2–norm.

Level l	$\|u - u_l^{L,ex}\|_{H^1}$	$\|u - u_l^{L,ex}\|_{L_2}$	$\|u - u_l^{Q}\|_{H^1}$	$\|u - u_l^{Q}\|_{L_2}$
3	0.5038-03	0.4353-05	0.5038-03	0.4354-05
4	0.1246-03	0.5269-06	0.1246-03	0.5269-06
5	0.3101-04	0.5861-07	0.3101-04	0.5858-07

Table 4: Comparison of the discretization errors

Conclusion

In this paper we have presented the analysis of an algorithm which can algebraically be understood as multigrid with τ-extrapolation. In practice, this algorithm is simple to implement, once a multigrid algorithm is available. However, we have shown that the algorithm converges to the same solution as a higher order fine element discretization. The algorithm can thus be used on unstructured meshes in an adaptive refinement setting. Furthermore, it is independent of global error expansions, and can thus be applied locally.

REFERENCES

[1] Marchuk, G. and Shaidurov, V., *Difference Methods and their Extrapolations*, Springer, New York, 1983.

[2] Blum, H., Lin, Q., and Rannacher, R., Asymptotic Error Expansions and Richardson Extrapolation for Linear Finite Elements, *Numer. Math.*, 49:11–37, 1986.

[3] Brandt, A., Multigrid Techniques: 1984 Guide with Applications to Fluid Dynamics, *GMD Studien*, 85, 1984.

[4] Hackbusch, W., *Multigrid Methods and Applications*, Springer Verlag, Berlin, 1985.

[5] Schaffer, S., Higher Order Multigrid Methods, *Math. Comp.*, 43:89–115, 1984.

[6] Bernert, K., Tauextrapolation – theoretische Grundlagen, numerische Experimente und Anwendung auf die Navier–Stokes–Gleichungen, Preprint SPC 94_7, Technische Universität Chemnitz–Zwickau, Fakultät für Mathematik, 1994.

[7] McCormick, S. and Rüde, U., On Local Refinement Higher Order Methods for Elliptic Partial Differential Equations, *International Journal of High Speed Computing*, 2(4):311–334, 1990, Also available as TU-Bericht I-9034.

[8] Jung, M. and Rüde, U., Implicit extrapolation methods for multilevel finite element computations: Theory and application, Preprint SPC 94_11, Technische Universität Chemnitz–Zwickau, Fakultät für Mathematik, 1994.

[9] Ciarlet, P., *The finite element method for elliptic problems*, North–Holland, Amsterdam, 1978.

[10] Steidten, T. and Jung, M., Das Multigrid–Programmsystem FEMGPM zur Lösung elliptischer und parabolischer Differentialgleichungen einschließlich mechanisch–thermisch gekoppelter Probleme (Version 06.90), Programmdokumentation, Technische Universität Karl–Marx–Stadt (Chemnitz–Zwickau), Sektion Mathematik, 1990.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	September 1996	Conference Publication

4. TITLE AND SUBTITLE

Seventh Copper Mountain Conference on Multigrid Methods

5. FUNDING NUMBERS

WU 505-59-53-01

6. AUTHOR(S)

N. Duane Melson, Tom A. Manteuffel, Steve F. McCormick, and Craig C. Douglas, Editors

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

NASA Langley Research Center
Hampton, VA 23681-0001

8. PERFORMING ORGANIZATION REPORT NUMBER

L-17593A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Washington, DC 20546-0001

Department of Energy
Washington, DC 20585

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

NASA CP-3339
Part 1

11. SUPPLEMENTARY NOTES

Organizing Institutions: University of Colorado at Denver; Front Range Scientific Computations, Inc.; and the Society for Industrial and Applied Mathematics

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified–Unlimited
Subject Category 64
Availability: NASA CASI (301) 621-0390

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The Seventh Copper Mountain Conference on Multigrid Methods was held on April 2–7, 1995 at Copper Mountain, Colorado. This book is a collection of many of the papers presented at the conference and so represents the conference proceedings. NASA Langley graciously provided printing of this document so that all of the papers could be presented in a single forum. Each paper was reviewed by a member of the conference organizing committee under the coordination of the editors.

The multigrid discipline continues to expand and mature, as is evident from these proceedings. The vibrancy in this field is amply expressed in these important papers, and the collection shows its rapid trend to further diversity and depth.

14. SUBJECT TERMS

Multigrid; Algorithms; Computational fluid dynamics (CFD)

15. NUMBER OF PAGES

424

16. PRICE CODE

A18

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	